

TUGAS AKHIR - TE 141599

**IMPLMENTATASI SENSOR KINECT PADA *MOBILE* ROBOT UNTUK
INSPEKSI OBJEK YANG MENGANDUNG BAHAN KIMIA**

Muhammad Iqbal Fahmi
NRP 07111645000035

Dosen Pembimbing
Dr. Muhammad Rivai, ST.MT
Dr. Ir. Hendra Kusuma, M.Eng.Sc

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya
2018



TUGAS AKHIR - TE 141599

**IMPLEMENTASI SENSOR KINECT PADA *MOBILE* ROBOT
UNTUK INSPEKSI OBJEK YANG MENGANDUNG BAHAN
KIMIA**

Muhammad Iqbal Fahmi
NRP 07111645000035

Dosen Pembimbing :
Dr. Muhammad Rivai, ST.MT
Dr. Ir. Hendra Kusuma, M.Eng.Sc

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018



FINAL PROJECT - TE 141599

IMPLEMENTATION KINECT SENSOR ON MOBILE ROBOT FOR INSPECTION OF CHEMICAL CONTAINERS

Muhammad Iqbal Fahmi
NRP 07111645000035

Advisor Lecturer :
Dr. Muhammad Rivai, ST.MT
Dr. Ir. Hendra Kusuma, M.Eng.Sc

ELECTRICAL ENGINEERING DEPARTEMENT
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2018

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Implementasi Sensor Kinect Pada *Mobile Robot* Untuk Inspeksi Objek Yang Mengandung Bahan Kimia**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juli 2018



Muhammad Iqbal Fahmi
NRP 07111645000035

**HALAMAN PENGESAHAN
IMPLEMENTASI SENSOR KINECT PADA MOBILE
ROBOT UNTUK INSPEKSI OBJEK YANG
MENGANDUNG BAHAN KIMIA**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Elektronika
Departemen Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui :

Dosen Pembimbing I

Dosen Pembimbing II

Dr. Muhammad Rivai, ST., MT.
NIP. 196904261994031003

Dr. Ir. Hendra Kusuma, M.Eng.Sc.
NIP. 196409021989031003



IMPLEMENTASI SENSOR KINECT PADA *MOBILE* ROBOT UNTUK INSPEKSI OBJEK YANG MENGANDUNG BAHAN KIMIA

Nama : Muhammad Iqbal Fahmi
Pembimbing I : Dr. Muhammad Rivai, ST.MT.
Pembimbing II : Dr. Ir. Hendra Kusuma, M.Eng.Sc.

ABSTRAK

Adanya teror bom yang terjadi saat ini dengan cara meletakkan barang-barang mencurigakan di lokasi tertentu masih banyak terjadi. Metode saat ini untuk menjinakkan bom menggunakan rompi anti-peledak khusus masih merupakan risiko yang berbahaya. Pada penelitian ini telah dibuat sebuah *Mobile Robot* yang dilengkapi Sensor Kinect, Modul Sensor Gas MQ-5, Metode HSV dan *Image Proccesing*. Pada pengujian yang mengimplementasi sensor kinect pada *mobile* robot untuk inspeksi warna objek target dan dilengkapi sensor gas MQ-5 untuk melakukan aktifitas pemetaan terhadap gas Iso-Butane pada suatu lokasi objek. Hasil percobaan *image proccesing* dengan kamera sensor kinect ketika mendapatkan nilai objek warna hijau dengan metode HSV adalah hue 29 - 84, saturation 86 - 255, value 6 - 255 dengan percahayaan ruangan 2000 lux, kesalahan nilai warna objek hijau dengan percahayaan ruangan 80 lux mengakibatkan mendeteksi warna objek kuning. Pada hasil pemetaan sensor gas menunjukan arah dan kecepatan angin melebihi 5 km/jam mempengaruhi kenaikan dan penurunan nilai pembacaan kadar gas, sedangkan ketika pengukuran jarak menggunakan depth sensor kinect antara jarak *mobile* robot dengan objek minimal 40 cm - 400 cm.

Kata Kunci : Metode HSV, Sensor Kinect, Sensor MQ-5

-----Halaman ini sengaja dikosongkan-----

IMPLEMENTATION KINECT SENSOR ON MOBILE ROBOT FOR INSPECTION OF CHEMICAL CONTAINERS

Name : Muhammad Iqbal Fahmi
Supervisor : Dr. Muhammad Rivai, ST.MT.
Co-Supervisor : Dr. Ir. Hendra Kusuma, M.Eng.Sc.

ABSTRACT

The existence of bombing terror that occurs at this time, by putting suspicious goods in a particular location is still a lot happening. The current method of taming a bomb using a special anti-explosive vest is still a dangerous risk. This research implements the Kinect sensor in a mobile robot for color inspection of the target object and equipped with an MQ-5 gas sensor to perform mapping activity on hazardous gas in an object location. Mobile robots are sent flat or indoor places. The result of camera Kinect sensor experiment when getting a value of a green color object with HSV method is hue 29 - 84, saturation 86 - 255, value 6 - 255 with room lighting 2000lux, green color object color error with room lighting 80lux. In the result of the gas, sensor mapping shows the direction and speed of wind exceeding 5 km/hour influence the increase and decrease of gas readout value, whereas when distance measurement uses Kinect depth sensor between mobile robot distance with object minimum 40 cm and maximum 4 cm.

Keywords : Method HSV, Sensor Kinect, Sensor MQ-5

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Alhamdulillahirobbil'alamin, puji syukur senantiasa panjatkan kepada Allah *Subhanahu wa Ta'ala* yang atas rahmat, petunjuk dan kasih sayang-Nya, sehingga penelitian dan penulisan tugas akhir dengan judul “Implementasi Sensor Kinect Pada *Mobile Robot* Untuk Inspeksi Objek Yang Mengandung Bahan Kimia” ini dapat diselesaikan dengan baik.

Dalam proses menyelesaikan tugas akhir ini penulis mendapat banyak bantuan dan dukungan dari berbagai pihak. Penulis mengucapkan banyak terima kasih pada:

1. Dr. Muhammad Rivai, ST.MT dan Dr. Ir. Hendra Kusuma, M.Eng.Sc selaku dosen pembimbing atas bimbingan dan arahan selama penulis mengerjakan tugas akhir ini.
2. Ir. Tasripan, M.T., Ir. Totok Mujiono, M.Kom, Astria Nur Irfansyah, ST., M.Eng, dan Fajar Budiman ST., M.Sc selaku dosen penguji atas pengujian tugas akhir dan arahan selama penulisan mengerjakan revisi buku tugas akhir ini.
3. Dr.Eng. Ardyono Priyadi, S.T., M.Eng. selaku Ketua Jurusan Teknik Elektro ITS Surabaya.
4. Seluruh dosen bidang studi Elektronika Departemen Teknik Elektro ITS Surabaya.
5. Bapak, Ibu, Kakak, Adik yang telah memberikan dukungan baik moril maupun materi.
6. Teman-teman yang senantiasa memberikan semangat dan motivasi kepada penulis.

Penulis menyadari bahwa tugas akhir ini tidaklah sempurna, namun penulis berharap tugas akhir ini dapat menjadi bahan bagi rekan-rekan untuk menambah wawasan serta dapat membantu pengerjaan tugas akhir selanjutnya.

Surabaya, Juli 2018

Muhammad Iqbal Fahmi

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

HALAMAN

PERNYATAAN KEASLIAN TUGAS AKHIR	Error! Bookmark not defined.
HALAMAN PENGESAHAN	Error! Bookmark not defined.
ABSTRAK	xi
ABSTRACT	xiii
KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxiii
 BAB I PENDAHULUAN	 1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	2
1.4 Batasan Masalah	2
1.5 Metodologi Penelitian	2
1.5.1 Studi Literatur	2
1.5.2 Perancangan Sistem	3
1.5.3 Pengujian Sistem	3
1.5.4 Pengolahan Data	3
1.5.5 Penulisan Laporan Tugas Akhir	3
1.6 Sistematika Laporan	3
1.7 Relevansi	4
 BAB II DASAR TEORI	 5
2.1 Gas Beracun dan Berbahaya	5
2.1.1 Karbon Monoksida (CO)	5
2.1.2 Metana (CH ₄)	6
2.1.3 Liquidified Petroleum Gas (LPG)	6
2.2 Sensor MQ-5	6
2.3 Pengolahan Citra Digital	9
2.3.1 Citra Dengan Ruang Warna RGB	9
2.3.2 Citra Dengan Ruang Warna HSV	10
2.3.3 Citra Biner	11
2.3.4 Filter Dilatasi dan Erosi	12
2.4 Sensor Kinect	13

2.5 OpenComputer Vision (OpenCV)	14
2.6 Python.....	15
2.7 Mikrokonroller Arduino Uno.....	15
2.8 Kendali Proposional Integral Derivatif	16
2.8.1 Kendali Proposional	17
2.9 <i>Driver</i> Motor VNH2SP30	17
BAB III PERANCANGAN SISTEM.....	21
3.1 Arsitektur Sistem.....	21
3.2 Desain Mekanik Robot	23
3.2.1 Perancangan Sensor Kinect.....	24
3.2.2 Perancangan Sensor MQ-5	25
2.9.1 Perancangan <i>Driver</i> Motor Dc VNH2SP30	27
3.3 Perancangan Perangkat Lunak	28
3.3.1 Mengakses Sensor Kinect.....	30
3.3.2 Mengakses HSV	30
3.3.3 Mengakses Filter Erosi dan Dilatasi	31
3.3.4 Mengakses Pengukuran Jarak	32
3.3.5 Mengakses Sensor MQ-5.....	32
BAB IV PENGUJIAN DAN PEMBAHASAN SISTEM	35
4.1 Hasil Realisasi Desain Mekanik <i>Mobile</i> Robot	35
4.2 Pengujian Pengolahan Citra	36
4.3 Pengujian Mendapatkan Nilai Warna HSV	37
4.4 Pengujian Pengukuran Jarak Menggunakan Depth Sensor Kinect...40	
4.5 Pengujian Pengukuran Jarak Objek Dengan Luasan Objek.....41	
4.6 Pengujian Sensor MQ-5	42
4.7 Pengujian Integrasi Sistem <i>Mobile</i> Robot	45
4.7.1 Posisi Objek di Depan <i>Mobile</i> Robot.....	45
4.7.2 Posisi Objek di Kanan <i>Mobile</i> Robot.....	46
4.7.3 Posisi Objek di Kiri <i>Mobile</i> Robot	48
BAB V PENUTUP.....	51
1.1 Kesimpulan	51
1.2 Saran	52
DAFTAR PUSTAKA.....	53
LAMPIRAN	55

DAFTAR GAMBAR

	HALAMAN
Gambar 2. 1 (a) Sensor MQ-5, dan (b) Modul sensor MQ-5	6
Gambar 2. 2 (1) Konfigurasi pin sensor, dan (2) Rangkaian sensor MQ-5	8
Gambar 2. 3 Karakteristik sensor MQ-5	8
Gambar 2. 4 Model ruang warna RGB	9
Gambar 2. 5 Citra dengan ruang warna RGB.....	10
Gambar 2. 6 Model ruang warna HSV	10
Gambar 2. 7 Citra dengan ruang warna HSV	11
Gambar 2. 8 Citra dengan biner	11
Gambar 2. 9 Dilatasi pada grayscale image	12
Gambar 2. 10 Erosi pada grayscale image	12
Gambar 2. 11 Filter dilatasi dan erosi	13
Gambar 2. 12 Komponen- komponen dari sensor Kinect	13
Gambar 2. 13 Blok diagram internal dari sensor Kinect	14
Gambar 2. 14 Board arduino uno	16
Gambar 2.15 Driver motor VNH2SP30.....	17
Gambar 2. 16 Skematik <i>driver</i> motor VNH2SP30	18
 Gambar 3. 1 Arsitektur sistem <i>mobile</i> robot	 21
Gambar 3. 2 Perancangan sistem perangkat keras elektronis.....	22
Gambar 3. 3 Desain mekanik tampak samping	23
Gambar 3. 4 Desain mekanik tampak atas	23
Gambar 3. 5 Algoritma <i>mobile</i> robot.....	24
Gambar 3. 6 Perancangan sensor kinect	24
Gambar 3. 7 Perancangan sensor MQ-5	25
Gambar 3. 8 Grafik perbandingan Rs/Ro dengan kadar gas	26
Gambar 3. 9 Grafik persamaan kadar ppm gas <i>iso-butane</i>	27
Gambar 3. 10 Perancangan driver motor dc VNH2SP30.....	27
Gambar 3. 11 Diagram algoritma <i>mobile</i> robot.....	28
Gambar 3. 12 Perancangan diagram <i>mobile</i> robot.....	28
Gambar 3. 13 Diagram alur sistem <i>image</i>	29
 Gambar 4. 1 Realisai desain sistem <i>mobile</i> robot.....	 35
Gambar 4. 2 Citra asli.....	36
Gambar 4. 3 Citra hasil pengubahan RGB ke HSV	36

Gambar 4. 4	Citra hasil filter dilatasi dan erosi	36
Gambar 4.5	Citra hasil proses perolehan fitur	36
Gambar 4. 6	Pengujian mencari masukan nilai HSV	37
Gambar 4. 7	Pengujian hasil mencari manual berwarna hijau dengan indikator percahayaan 2000 lux	37
Gambar 4. 8	Hasil pengujian percahayaan lux meter dengan menggunakan aplikasi handphone	37
Gambar 4. 9	Pengujian hasil mencari manual berwarna hijau dengan indikator percahayaan 225 lux	38
Gambar 4. 10	Hasil pengujian percahayaan lux meter dengan menggunakan aplikasi handphone	38
Gambar 4. 11	Pengujian hasil mencari manual berwarna hijau berubah deteksi menjadi kuning akibat percahayaan 80 lux	38
Gambar 4. 12	Hasil pengujian percahayaan lux meter dengan menggunakan aplikasi handphone	39
Gambar 4. 13	Implementasi pengujian sensor depth pada objek	40
Gambar 4. 14	Grafik pengujian jarak menggunakan depth sensor kinect.....	40
Gambar 4. 15	Interface depth sensor.....	41
Gambar 4. 16	Jarak sensor kinect pada <i>mobile</i> robot dengan objek.....	41
Gambar 4. 17	Pengujian <i>mobile</i> robot dengan gas <i>iso-butane</i>	43
Gambar 4. 18	Grafik hasil data sensor gas MQ-5 ppm terhadap waktu.....	43
Gambar 4. 19	Grafik hasil data sensor gas MQ-5 Rs/Ro terhadap waktu.....	43
Gambar 4. 20	Algoritma sistem <i>mobile</i> robot pada posisi objek di depan	45
Gambar 4. 21	Hasil implementasi algoritma sistem <i>mobile</i> robot pada posisi objek di depan	45
Gambar 4. 22	Citra hasil filter dilatasi dan erosi dan citra hasil proses perolehan fitur tampak depan.....	46
Gambar 4. 23	Hasil data error dari deteksi objek tampak depan	46
Gambar 4. 24	Algoritma sistem <i>mobile</i> robot pada posisi objek di kanan	46
Gambar 4. 25	Hasil implementasi algoritma sistem <i>mobile</i> robot pada posisi objek di kanan.....	47
Gambar 4. 26	Citra hasil filter dilatasi dan erosi dan citra hasil proses perolehan fitur tampak kanan.....	47

Gambar 4. 27	Hasil data error dari deteksi objek tampak kanan	48
Gambar 4. 28	Algoritma sistem <i>mobile</i> robot pada posisi objek di kiri	48
Gambar 4. 29	Hasil implementasi algoritma sistem <i>mobile</i> robot pada posisi objek di kiri	49
Gambar 4. 30	Citra hasil filter dilatasi dan erosi dan citra hasil proses perolehan fitur tampak kiri	49
Gambar 4. 31	Hasil data error dari deteksi objek tampak kiri	49

-----Halaman ini sengaja dikosongkan-----

DAFTAR TABEL

	HALAMAN
Tabel 2. 1 Keterangan modul sensor gas MQ-5	7
Tabel 2. 2 Karakteristik sensor MQ-5	7
Tabel 2. 3 Karakteristik arduino uno	16
Tabel 2. 4 Keterangan driver motor VNH2SP30	18
Tabel 2. 5 Spesifikasi motor driver VNH2SP30	19
 Tabel 3. 1 Nilai ppm gas terhadap perbandingan Ro/Rs	 26
 Tabel 4. 1 Hasil mencari nilai color HSV	 39
Tabel 4. 2 Hasil data pengukuran jarak Objek dengan <i>mobile</i> robot	41
Tabel 4. 3 Hasil data pemberian gas iso-butane	44

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada era perkembangan teknologi saat ini tingkat kriminalitas dan terorisme semakin meningkat. Pelaku terorisme menggunakan kemajuan teknologi untuk mengembangkan aksi teror seperti pengeboman. Berbagai macam bom yang dirakit banyak menggunakan bahan kimia berbahaya. Bahan kimia yang membentuk zat berbahaya sangat mudah didapat dan telah digunakan pada kasus terorisme di berbagai negara.

Berdasarkan data yang diperoleh dari sebuah sumber pada bulan September 2015 menunjukkan bahwa penindakan yang dilakukan kepada pelaku tindak pidana terorisme sampai dengan September 2015 berjumlah 1.143 orang. Dari jumlah tersebut yang sudah bebas sebanyak 501 orang, yang masih menjalani hukuman sebanyak 328 orang, dan tersangka yang dikembalikan kepada keluarga sebanyak 98 orang. Selain itu ada tersangka yang meninggal dunia sebanyak 127 orang, meninggal dunia karena penegakkan hukum sebanyak 108, meninggal dunia karena eksekusi mati sebanyak 3 orang, dan mati karena bunuh diri sebanyak 16 orang [1].

Sebagai upaya untuk mencegah terjadi kecelakaan dalam kegiatan penjinakan bom salah satu kemajuan teknologi terbaru saat ini adalah penggunaan *mobile* robot yang mampu menggantikan tugas manusia dalam melakukan pekerjaan yang berbahaya, dalam bidang keamanan dengan tujuan untuk mendeteksi suatu objek yang mengandung bahan kimia dari benda yang mencurigakan, sehingga resiko terjadinya kecelakaan pada manusia dapat berkurang.

Mobile robot yang sudah ada memiliki beberapa kelemahan dalam hal keakuratan mendekati target dan beberapa robot penjinak bom belum terpasang kamera yang dapat berfungsi sebagai sensor untuk mendekati target [2] [3] . Belum dilengkapi sensor gas semikonduktor untuk menginspeksi kandungan gas pada target.

Menyadari hal tersebut maka diusulkan sebuah *mobile* robot yang dapat berjalan diarea yang datar, mendekati target objek menggunakan sensor kinect dan mendeteksi kandung kadar gas *iso-butane* dengan sensor gas MQ-5.

1.2 Rumusan Masalah

Berdasar latar belakang diatas, dapat dirumuskan masalah sebagai berikut:

1. Bagaimana cara mengidentifikasi objek target dengan menggunakan sensor kinect.
2. Cara untuk berhenti antara jarak *mobile* robot terhadap target objek dengan sensor kinect.
3. Bagaimana cara menginspeksi kandungan kadar gas *iso-butane* pada target objek dengan menggunakan modul sensor MQ-5.

1.3 Tujuan Penelitian

Penelitian pada tugas akhir ini bertujuan sebagai berikut:

1. Mampu Mengenali objek dengan warna dengan menggunakan sensor kinect.
2. Implementasi sensor kinect dalam berhentinya *mobile* robot dengan target objek.
3. Pengukuran kandungan konsentrasi gas menggunakan modul sensor gas MQ-5.

1.4 Batasan Masalah

1. *Mobile* Robot dijalankan pada tempat yang datar, tidak terdapat halangan dan tidak tergenang air.
2. *Mobile* robot dapat mendeteksi objek target sesuai dengan masukan nilai jenis warna yang sudah ditentukan.
3. Sensor kinect dapat mengukur jarak berhentinya *mobile* robot dengan target objek dibawah 4 meter.
4. Gas *iso-butane* dijadikan sebagai bahan uji gas berbahaya.

1.5 Metodologi Penelitian

Dalam penyelesaian tugas akhir ini digunakan metodologi sebagai berikut :

1.5.1 Studi Literatur

Pada tahap studi literatur dilakukan pengumpulan dasar teori yang menunjang dalam penulisan tugas akhir. Berikut studi literatur yang dilakukan :

- a. Studi tentang protokol sensor kinect.
- b. Studi tentang metode pengolahan citra HSV.
- c. Studi tentang akusisi data pada sensor gas MQ-5.

Dasar teori ini dapat diambil dari buku-buku jurnal, proceeding, dan artikel-artikel di internet.

1.5.2 Perancangan Sistem

Setelah mempelajari literatur yang ada, selanjutnya akan dilakukan perancangan sistem. Sistem yang akan dirancang meliputi perancangan desain mekanik *mobile* robot, integrasi modul-modul sensor kinect, sensor gas dan *driver* motor dc.

1.5.3 Pengujian Sistem

Pengujian sistem dilakukan untuk menentukan keandalan dari sistem yang telah dibuat. Pengujian dilakukan untuk melihat apakah perangkat keras dan perangkat lunak telah bekerja. Dilakukan pada setiap sensor yaitu sensor kinect dan sensor MQ-5. Sensor kinect untuk menguji dalam mendeteksi objek dan mencari nilai HSV dari jenis warna objek. Pengujian sensor gas dilakukan dengan memberikan gas *iso-butane* di area sekitaran sensor gas dan kemudian merekam hasilnya. Pengujian keakuratan integrasi sistem *mobile* robot dalam mendekati objek target dan berhenti *mobile* robot.

1.5.4 Pengolahan Data

Data yang telah diperoleh dari sensor kinect diakuisisi oleh pc dan modul sensor gas MQ-5 yang diakuisisi oleh mikrokontroler pada minimum sistem yaitu arduino uno. Data dari sensor kinect yang diakuisisi oleh pc untuk mendeteksi warna objek target dan diolah untuk digunakan sebagai acuan tujuan dan data dari sensor gas diolah untuk dapat diidentifikasi nilai kandungan kadar gas.

1.5.5 Penulisan Laporan Tugas Akhir

Tahap penulisan laporan Tugas Akhir dilakukan pada saat pengujian sistem dimulai dan setelahnya.

1.6 Sistematika Laporan

Pembahasan tugas akhir ini dibagi menjadi lima bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Pada bab pendahuluan, menjelaskan mengenai latar belakang pemilihan topik, perumusan masalah dan batasannya. Bab ini juga membahas mengenai tujuan penelitian, metodologi, sistematika laporan, dan relevansi dari penelitian yang dilakukan.

Bab II Dasar Teori

Penjelasan mengenai perangkat keras maupun perangkat lunak pendukung implementasi sensor kinect pada *mobile* robot untuk inspeksi objek yang mengandung bahan kimia dibahas pada bab ini. Pokok bahasan pada bab ini diantaranya opencv, python, sensor kinect, sensor gas MQ-5 , driver motor dc VNH2SP30 spesifikasi dan cara kerja.

Bab III Perancangan Sistem

Pembahasan yang dilakukan pada bab ini, mengenai perancangan sistem secara keseluruhan serta prosedur koneksi perangkat keras dan perangkat lunak yang dibuat.

Bab IV Pengujian dan Pembahasan sistem

Hasil dari uji coba program yang dibuat serta perangkat keras *mobile* robot dibahas secara lengkap pada bab ini. Bab ini menjelaskan tentang hasil yang didapat dalam pengujian dari tiap blok sistem dan subsistem serta hasil evaluasi system tersebut.

Bab V Penutup

Bab ini menjelaskan tentang kesimpulan meliputi hasil yang dicapai, dan kekurangan-kekurangan pada kerja alat dari hasil analisa serta saran untuk pengembangan ke depan.

1.7 Relevansi

Hasil yang diharapkan dari tugas akhir ini diharapkan mampu meningkatkan keamanan teror barang dan keselamatan bagi manusia sendiri dengan menggunakan *mobile* robot ini.

BAB II

DASAR TEORI

2.1 Gas Beracun dan Berbahaya

Terdapat beberapa macam gas pengotor dalam udara bebas atau tambang bawah tanah dapat digolongkan menjadi gas beracun dan gas berbahaya. Gas-gas ini berasal dari proses-proses yang terjadi dalam tambang, batuan ataupun bahan galiannya, maupun berasal dari proses aktifitas manusia. Gas-gas yang bersifat gas beracun adalah gas yang bereaksi dengan darah dan dapat menyebabkan kematian. Sedangkan gas berbahaya adalah gas-gas pengotor yang menyebabkan bahaya, baik terhadap kehidupan manusia maupun dapat menyebabkan peledakan.

Beberapa gas yang tergolong dalam gas beracun adalah :

2.1.1 Karbon Monoksida (CO)

Gas karbon monoksida merupakan gas yang tidak berwarna, tidak berbau dan tidak ada rasa, dapat terbakar dan sangat beracun. Gas ini banyak dihasilkan dari sisa pembakaran senyawa hidrokarbon. Gas ini mempunyai afinitas yang tinggi terhadap haemoglobin darah, sehingga sedikit saja kandungan gas CO dalam udara akan segera bersenyawa dengan butir-butir haemoglobin (COHb) yang akan meracuni tubuh lewat darah. Afinitas CO terhadap haemoglobin menurut penelitian (Forbes and Grove, 1954) mempunyai kekuatan 300 kali lebih besar dari pada oksigen dengan hemoglobin. Gas CO dihasilkan dari hasil pembakaran, operasi motor bakar, proses peledakan dan oksidasi lapisan batubara [4].

Karbon monoksida merupakan gas beracun yang sangat mematikan karena sifatnya yang kumulatif. Misalnya gas CO pada kandungan 0,04% dalam udara apabila terhirup selama satu jam baru memberikan sedikit perasaan tidak enak, namun dalam waktu 2 jam dapat menyebabkan rasa pusing dan setelah 3 jam akan menyebabkan pingsan/ tidak sadarkan diri dan pada waktu lewat 5 jam dapat menyebabkan kematian. Kandungan CO sering juga dinyatakan dalam ppm (part per milion). Sumber CO yang sering menyebabkan kematian adalah gas buangan dari mobil dan kadang-kadang juga gas pemanas air. Gas CO mempunyai berat jenis 0,9672 sehingga selalu terapung dalam udara.

2.1.2 Metana (CH₄)

Gas metana ini merupakan gas yang selalu berada dalam tambang batubara dan sering merupakan sumber dari suatu peledakan tambang. Campuran gas metana dengan udara disebut '*Firedamp*'. Apabila kandungan metana dalam udara tambang bawah tanah mencapai 1% maka seluruh hubungan mesin listrik harus dimatikan. Gas ini mempunyai berat jenis yang lebih kecil dari pada udara dan karenanya selalu berada pada bagian atas dari jalan udara. Metana merupakan gas yang tidak beracun, tidak berwarna, tidak berbau dan tidak mempunyai rasa. Pada saat proses pembatubaraan terjadi maka gas metana terbentuk bersama-sama dengan gas karbondioksida. Gas metana ini akan tetap berada dalam lapisan batubara selama tidak ada perubahan tekanan padanya.

Terbebasnya gas metana dari suatu lapisan batubara dapat dinyatakan dalam suatu volume per satuan luas lapisan batubara, tetapi dapat juga dinyatakan dalam satuan volume per satuan waktu. Terhadap kandungan gas metana yang masih terperangkap dalam suatu lapisan batu bara dapat dilakukan penyedotan dari gas metana tersebut dengan pompa untuk dimanfaatkan. Proyek ini dikenal dengan nama '*seam methane drainage*' [4].

2.1.3 Liquidified Petroleum Gas(LPG)

Liquidified Petroleum Gas (LPG), merupakan gas minyak bumi yang dicairkan. Dengan menambah tekanan dan menurunkan suhunya, gas berubah menjadi cair. Komponennya didominasi propana (C₃H₈) dan butane (C₄H₁₀). Elpiji juga mengandung hidrokarbon ringan lain dalam jumlah kecil, misalnya etana (C₂H₆) dan pentana (C₅H₁₂). Dalam kondisi atmosfer, elpiji akan berbentuk gas. Volume elpiji dalam bentuk cair lebih kecil dibandingkan dalam bentuk gas untuk berat yang sama. Gas ini mempunyai sifat sangat mudah terbakar, mempunyai masa yang lebih berat dari udara, tidak beracun, tidak berwarna, tetapi berbau menyengat.

2.2 Sensor MQ-5



Gambar 2. 1 (a) Sensor MQ-5, dan (b) Modul sensor MQ-5

Pada Gambar 2.1 adalah jenis modul sensor gas MQ-5 dengan keterangan pada Tabel 2.1. Sensor ini memang tidak sesensitif sensor gas lainnya, namun kelebihanannya adalah sifatnya yang universal yang mampu mendeteksi tipe gas yang lebih luas. Pada Gambar 2.1.a di atas ini merupakan gambar sensor MQ-5 sedangkan Gambar 2.1.b. merupakan gambar sensor MQ-5 yang telah dikemas menjadi sebuah modul. Sensor MQ-5 adalah sensor universal yang mampu mendeteksi berbagai jenis gas seperti hidrogen (H_2), karbon monoksida (CO), metana (CH_4), etanol (CH_3CH_2OH), propana (C_3H_8), butane (C_4H_{10}), dan gas hidrokarbon lainnya.

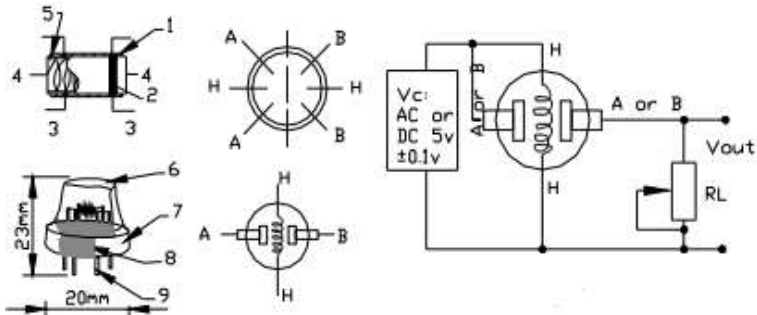
Tabel 2. 1 Keterangan modul sensor gas MQ-5

No	Pin	Keterangan
1	A Out	Analog keluaran
2	D Out	Digital keluaran
3	GND	GND
4	VCC	Tegangan pengoperasian

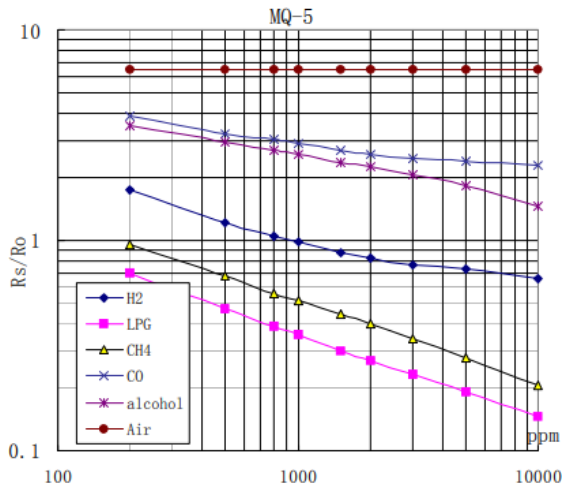
Secara umum struktur dari sensor MQ-5 seperti pada Gambar 2.2 dibawah ini, gambar (1) menunjukkan konfigurasi pin sensor yang keterangannya dapat dilihat pada Tabel 2.1 dan gambar (2) menunjukkan rangkaian sensor MQ-5 dengan tambahan trimpot dan pin header :

Tabel 2. 2 Karakteristik sensor MQ-5

No	Nama	Material
1	Lapisan pendeteksi	SnO_2
2	Elektroda	Au
3	Kawat elektroda	Pt
4	Koil pemanas	Campuran Ni-Cr
5	Tabung keramik	Al_2O_3
6	Jaring anti ledakan	Stainless steel Gauze
7	Cincin penjepit	Plat tembaga
8	Dasar resin	Bakelite
9	Pin Konector	Plat tembaga



Gambar 2. 2 (1) Konfigurasi pin sensor, dan (2) Rangkaian sensor MQ-5 [5]



Gambar 2. 3 Karakteristik sensor MQ-5 [5]

Keterangan :

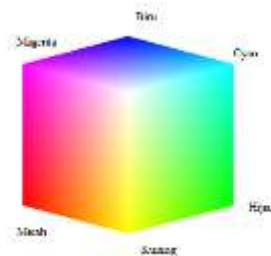
R_s : Resistansi sensor pada konsentrasi tertentu

R_o : Resistansi sensor pada 1000 ppm H_2 udara terbuka

Gambar 2.3 diatas ini merepresentasikan karakteristik sensitifitas dari sensor MQ-5 dan karakteristik ketergantungan temperatur dan kelembaban. Sumbu-y mengindikasikan sebagai rasio resistansi sensor yang ditentukan. Merupakan perbandingan resistansi (R_s/R_o) di mana R_s merupakan resistansi sensor ketika mendeteksi gas, dan R_o merupakan resistansi sensor pada udara bersih.

2.3 Pengolahan Citra Digital

Citra dapat didefinisikan sebagai fungsi dua dimensi, $f(x, y)$, di mana x dan y adalah koordinat bidang, dan amplitudo dan f pada setiap koordinat (x, y) , disebut intensitas. Saat semua nilai x , y dan amplitudo dari f memiliki nilai yang terbatas, yang bernilai diskrit, citra tersebut dengan citra digital. Pemrosesan citra digital dapat didefinisikan sebagai pemrosesan citra digital menggunakan computer digital. Citra digital tersusun atas elemen-elemen dengan jumlah tertentu yang terbatas, dimana masing-masing memiliki lokasi dan nilai tertentu. Elemen-elemen ini yang disebut dengan elemen citra, atau biasa disebut dengan pixel [6].



Gambar 2. 4 Model ruang warna RGB [6]

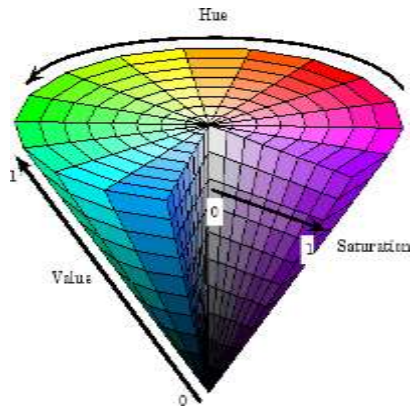
2.3.1 Citra Dengan Ruang Warna RGB

Berdasarkan penelitian, secara umum mata manusia dapat membedakan warna merah, hijau dan biru (Red, Green, Blue). Oleh karena itu, warna dapat dipandang sebagai kombinasi dari ketiga warna tersebut [6]. Ruang warna RGB ini biasa diterapkan pada kebanyakan sistem grafika computer. Ruang warna ini biasanya dimodelkan dalam bentuk kubus tiga dimensi, dengan warna merah, hijau dan biru berada pada sudut sumbu. Warna hitam berada pada titik asal dan warna putih berada di ujung kubus yang berseberangan. Gambar 2.4 menunjukkan visualisasi ruang warna RGB [6]. Pada Gambar 2.5 contoh citra dengan ruang warna RGB dengan warna hijau hasil dari kamera sensor kinect.

Pada aplikasinya, ruang warna ini tidak ideal untuk digunakan, karena warna merah, hijau dan biru sesungguhnya terkorelasi erat sehingga sulit untuk beberapa algoritma pemrosesan citra. Dalam aplikasi yang memproses warna tertentu, lebih mudah apabila menggunakan ruang warna HSV [6].



Gambar 2. 5 Citra dengan ruang warna RGB



Gambar 2. 6 Model ruang warna HSV [6]

2.3.2 Citra Dengan Ruang Warna HSV

HSV merupakan ruang warna yang merepresentasikan warna seperti yang dilihat oleh mata manusia. H berasal dari kata “hue”, yaitu sesuatu yang biasa disebut oleh sebagian orang sebagai warna, misalnya perbedaan antara warna merah dan merah muda. S berasal dari “saturation” yang menunjukkan jumlah warna yang ada. V berasal dari “value” yang menunjukkan jumlah cahaya yang ada, misalnya perbedaan antara merah gelap dengan merah cerah [7]. Model ruang warna HSV ditunjukkan oleh Gambar 2.6. Pada Gambar 2.7 contoh dari citra dengan ruang warna HSV dari pembacaan sensor kinect.

Nilai H, S dan V dapat diperoleh berdasarkan nilai R, G dan B dengan rumus sebagai berikut [8]:



Gambar 2. 7 Citra dengan ruang warna HSV

$$S = 1 - \frac{\min(R, G, B)}{V} \quad (2.1)$$

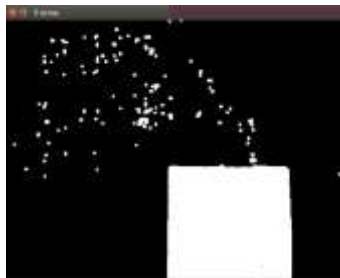
$$V = \frac{(R + G + B)}{3} \quad (2.2)$$

$$H = \tan\left(\frac{3(G-B)}{(R-G)+(R-B)}\right) \quad (2.3)$$

2.3.3 Citra Biner

Citra biner adalah citra dengan nilai piksel yang hanya memiliki dua kemungkinan, yaitu nilai 0, yang merepresentasikan warna hitam, atau 1, yang merepresentasikan warna putih. Citra jenis ini biasa digunakan untuk kepentingan memperoleh bentuk suatu objek, seperti yang dicontohkan pada Gambar 2.8.

Menkonversi suatu citra dengan ruang warna tertentu menjadi citra biner, digunakan suatu nilai yang disebut sebagai nilai ambang. Nilai tersebut dipakai untuk menentukan suatu intensitas akan dikonversi menjadi 1 [9].



Gambar 2. 8 Citra dengan biner

2.3.4 Filter Dilatasi dan Erosi

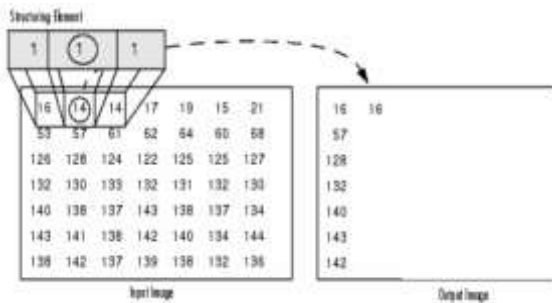
Dilatasi adalah proses penambahan pixel disekitar area pada gambar objek. Sedangkan erosi adalah proses pemindahan pixel untuk diletakkan disekitar objek. Jumlah pixel yang ditambahkan dan dipindahkan tergantung pada ukuran dan bentuk gambar dari struktur elemen yang digunakan pada image prosesing.

Ada dilatasi dan erosi memiliki aturan :

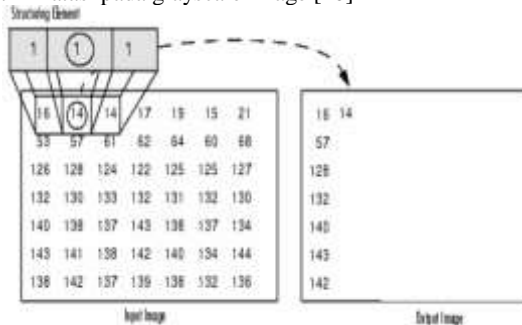
Dilatasi : Nilai keluaran pixel adalah maksimum pada semua nilai yang masukan. Pada gambar biner, apabila nilai masukan 1, maka hasil keluarannya juga 1 [10]. Yang tertera pada Gambar 2.9.

Erosi : Nilai output pixel adalah minimum pada semua nilai yang masukan. Pada gambar biner, apabila nilai masukan 0, maka hasil keluarannya juga 0 [10]. Yang tertera pada Gambar 2.10.

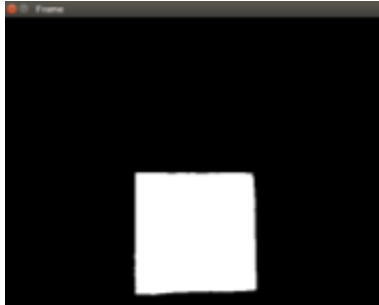
Contoh metode dilatasi dan erosi pada biner pada Gambar 2.11 contoh hasil filter dilatasi dan erosi :



Gambar 2. 9 Dilatasi pada grayscale image [10]



Gambar 2. 10 Erosi pada grayscale image [10]



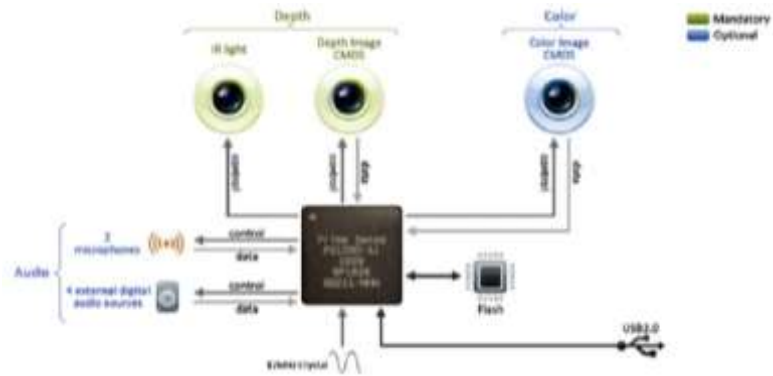
Gambar 2. 11 Filter dilatasi dan erosi

2.4 Sensor Kinect



Gambar 2. 12 Komponen- komponen dari sensor Kinect [11]

Sensor Kinect adalah sebuah aksesoris untuk *platform* game konsol Xbox 360. Kinect merupakan sebuah *motion sensing unit device* yang dikembangkan untuk mengintegrasikan gerak tubuh manusia. Kinect memiliki sensor-sensor yang terhubung dengan *motorized pivot based*. Perangkat ini memiliki kamera RGB , depth sensor, dan *multi-array microphone* yang terdapat pada Gambar 2.12. Dengan sensor-sensor tersebut Kinect dapat menyajikan kemampuan melakukan *full-body 3D motion capture*, *facial recognition*, dan *voice recognition*. Kinect memiliki performasi yang baik, dimensi yang kecil, *open source libraries* yang mampu membuat Kinect digunakan pada beberapa *platform OS* (Windows, Linux, dan Mac), serta memiliki harga yang cukup murah [11]. Sensor Microsoft Kinect mampu mengeluarkan keluaran video dengan *framerate* 60 Hz sebagai kombinasi dari dua sensor yang berbeda, pertama dihasilkan oleh kamera RGB dengan resolusi 8 bit VGA (640x480 pixel) dengan *Bayer Color Filter*, yang kedua dihasilkan oleh *monochrome depth sensor* dengan resolusi yang sama, dan 2048 tingkat sensitivitas (11 bit) [11].



Gambar 2. 13 Blok diagram internal dari sensor Kinect [11]

Bagaimana sensor-sensor pada Microsoft Kinect terhubung ditunjukkan pada Gambar 2.13 Sistem tersebut terdiri dari *depth camera* dan *color camera* yang menggunakan teknologi CMOS serta *microphone* sebagai sensor audio [11].

2.5 OpenComputer Vision (OpenCV)

OpenCV (*Open Source Computer Vision*) adalah sebuah API (*Application Programming Interface*) Library yang sudah sangat familiar pada pengolahan citra *Computer Vision*. *Computer Vision* itu sendiri adalah salah satu cabang dari Bidang Ilmu Pengolahan Citra (*Image Processing*) yang memungkinkan komputer dapat melihat seperti manusia. Dengan *vision* tersebut komputer dapat mengambil keputusan, melakukan aksi, dan mengenali terhadap suatu objek [12].

OpenCV dapat diterapkan pada bahasa pemrograman seperti C++, C, Python, Java, dan MATLAB. Sedangkan untuk sistem operasi, OpenCV mendukung untuk banyak sistem operasi seperti Windows, Linux, Android, dan Mac OS. Terdapat lebih dari 2500 algoritma yang telah dioptimalkan untuk melakukan algoritma *computer vision* dan *machine learning*. Beberapa pengimplementasian dari algoritma-algoritma tersebut adalah *Face Recognition*, *Face Detection*, *Face/Object Tracking*, *Road Tracking*, mengklasifikasi tindakan manusia dalam video, mengekstrak model-model 3D obyek, menghasilkan titik awan 3D dari kamera *stereo*, dan lain sebagainya [13].

2.6 Python

Python adalah sebuah bahasa pemrograman yang mudah dipelajari dan ampuh. Python memiliki struktur data tingkat tinggi yang efisien dan pendekatan terhadap pemrograman berorientasi objek yang sederhana namun efektif. Sintak python yang elegan dan typing dinamika, bersama-sama dengan sifatnya yang terinterpretir, menjadikan python bahasa yang ideal untuk pemrograman *scripting* dan *rapid application development* dalam berbagai bidang dan hampir semua *platform*. Python menjadi bahasa *official* yang terintegrasi dalam Raspberry Pi. Kata “Pi” pada Raspberry Pi merupakan singkatan yang merujuk pada “Python” [13].

Sisi utama yang membedakan Python dengan bahasa lain adalah dalam hal aturan penulisan kode program dengan aturan indentasi, tipe data, tuple, dan *dictionary*. Python memiliki kelebihan tersendiri dibandingkan dengan bahasa lain terutama dalam hal penanganan modul, ini yang membuat beberapa *programmer* menyukai python. Selain itu, python merupakan salah satu produk yang *open source*, *free*, dan *multi-platform* [13].

2.7 Mikrokontroler Arduino Uno

Arduino merupakan sebuah platform dari *physical computing* yang bersifat *opensource* dapat dilihat pada Gambar 2.14. Arduino adalah kombinasi dari perangkat keras, bahasa pemrograman, dan *Integrated Development Environment* (IDE) yang canggih. IDE adalah sebuah software yang sangat berperan untuk menulis program, *compile* menjadi kode biner dan *upload* ke dalam memori mikrokontroler [14].

Arduino Uno adalah board berbasis mikrokontroler pada ATmega328. Board ini memiliki 14 digital keluaran/masukan pin (dimana 6 pin dapat digunakan sebagai keluaran PWM), 6 masukan analog, 16 MHz osilator kristal, koneksi USB, jack listrik tombol reset. Pin-pin ini berisi semua yang diperlukan untuk mendukung mikrokontroler, hanya terhubung ke komputer dengan kabel USB atau sumber tegangan bisa didapat dari adaptor AC-DC atau baterai untuk menggunakannya [14].



Gambar 2. 14 Board arduino uno [15]

Tabel 2. 3 Karakteristik arduino uno

Mikrokontroler	Atmega 328
Tegangan pengoperasian	5V
Tegangan masukan yang disarankan	7-12V
Batas tegangan masukan	6-20V
Jumlah pin I/O digital	14 pin digital (6 diantaranya menyediakan keluaran PWM)
Jumlah pin masukan analog	6 pin
Arus DC tiap pin I/O	40mA
Arus DC untuk pin 3.3V	50mA
Memory flash	32 KB (Atmega 328)sekitar 0,5 KB digunakan oleh bootloader
SRAM	2 KB (Atmega 328)
EPROM	1 KB (Atmega 328)
Clock speed	16 MHz

2.8 Kendali Proporsional Integral Derivatif

Kendali proporsional integral derivatif adalah suatu sistem kendali yang merupakan gabungan dari kendali proporsional, kendali integral dan kendali derivatif dimana kendali proporsional akan menghasilkan keluaran kendali yang sebanding dengan nilai error sehingga diperoleh nilai steady state [16].

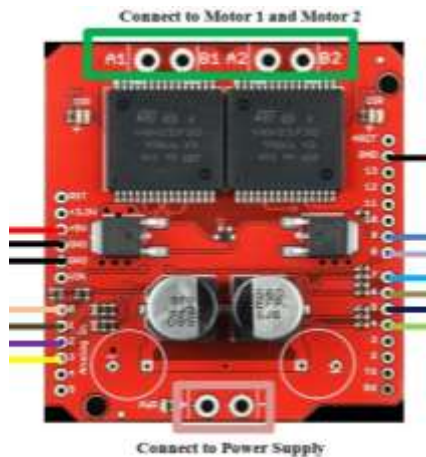
2.8.1 Kendali Proposional

Kendali proposional adalah kendali yang menghasilkan keluaran kendali yang sebanding dengan *error* masukan yang dikalikan dengan konstanta proposional. Kendali ini berfungsi untuk memperkuat sinyal *error*, sehingga mempercepat keluaran sistem mencapai titik referensi. Persamaan dari kendali proposional ini ditunjukkan pada persamaan 2.4

$$u(t) = K_p e(t) \quad (2.4)$$

2.9 Driver Motor VN12SP30

Merupakan H-Bridge berbasis VN12SP30 yang didisain untuk menghasilkan 2 arah dengan arus kontinyu sampai dengan 30 A pada tegangan 5,5 Volt sampai 36 Volt (IC VN12SP30 hanya sampai 16 V). Modul ini dilengkapi dengan rangkaian sensor arus beban yang dapat digunakan sebagai umpan balik ke pengendali. Modul ini mampu men-beban-beban induktif seperti misalnya relay, solenoida, motor DC, motor stepper, dan berbagai macam beban lainnya [17].



Gambar 2.15 Driver motor VN12SP30 [17]

Tabel 2. 5 Spesifikasi motor driver VN2SP30

Symbol	Parameter	Value
VCC	Supply voltage	+41 V
I _{max}	Maximum output current	30 A
I _R	Reverse output current	-30 A
I _{IN}	Input current(I _{IN_A} and I _{IN_B}) pins	±10mA
I _{EN}	Enable input current (DIAG _A /EN _A and DIAG _B /EN _B pins)	±10mA
I _{PW}	PWM input current	±10mA
V _{CS}	Current sense maximum voltage	-3/+15 V
V _{ESD}	Electro static discharger (R=1.5Kohm, C=100pF) - CS pin - Logic pin - Output pins: OUTA, OUTB, VCC	2kV 4kV 5kV
T _J	Junction operating temperature	Internally limited °C
T _C	Case operating temperature	-40 to 150 °C
T _{STG}	Storage operating temperature	-55 to 150 °C

-----Halaman ini sengaja dikosongkan-----

BAB III

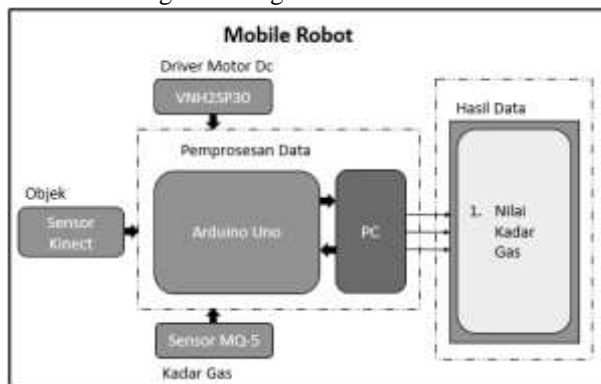
PERANCANGAN SISTEM

Pada bab ini menjelaskan tentang perancangan sistem yang meliputi arsitektur sistem, perancangan perangkat keras, perancangan perangkat lunak dan persiapan *mobile robot*.

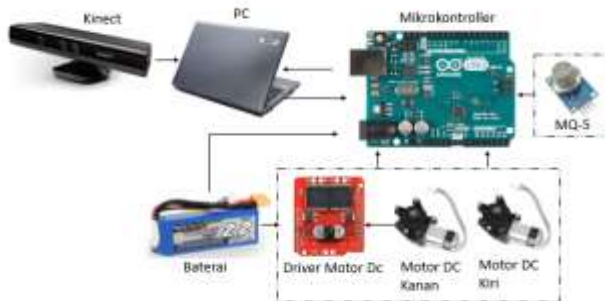
3.1 Arsitektur Sistem

Blok diagram dari gambar 3.1 merupakan penyederhanaan dari sistem *mobile robot* untuk mengidentifikasi objek yang mengandung kadar gas *iso-butane* berbahaya dengan memanfaatkan sensor kinect untuk mengenali objek warna, mengetahui kandungan kadar gas menggunakan sensor MQ-5 dan driver motor dc untuk mengontrol kecepatan roda sesuai dari data yang dikirim oleh sensor kinect dalam pembagian perbandingan pixel yang dikirimnya sebagai Tugas Akhir ini. Perancangan sistem ini bertujuan dapat berjalan sesuai dengan perencanaan. Yaitu dapat berjalan sesuai dengan arah objek dari masukan nilai HSV untuk dapat mengenali objek warna yang kemudian akan berjalan menuju objek, berhenti di depan objek dan mendeteksi nilai kandungan kadar gas.

Mempunyai antarmuka pada pc yang mana digunakan untuk mengetahui citra RGB dan citra hsv, citra sebelum dan sesudah dilakukan proses filter erosi dan dilatasi, citra hasil proses perolehan fitur dan nilai kandungan kadar gas.



Gambar 3. 1 Arsitektur sistem *mobile robot*



Gambar 3. 2 Perancangan sistem perangkat keras elektronik

Pada bagian Gambar 3.2 *mobile* robot, terdapat beberapa bagian seperti, sensor, Pemrosesan data, dan kendali. Pada bagian sensor, terdapat dua bagian komponen yaitu sensor kinect untuk mengetahui warna objek yang akan dijadikan target dan sensor MQ-5 untuk mengetahui nilai kandungan kadar gas. Pada bagian pemrosesan data terdapat mikrokontroler untuk mengolah data masukan dari sensor dan pc, yang nantinya akan digunakan untuk mengendalikan bagian kendali. Di mana bagian kendali ini terdapat aktuator yang berupa motor dc.

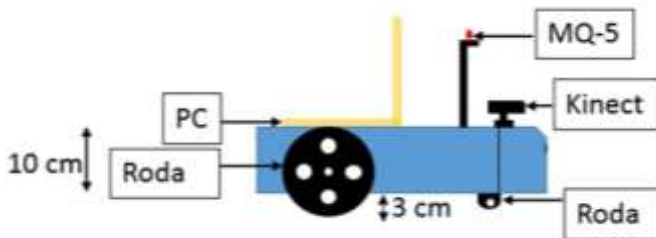
Secara keseluruhan sistem ini bekerja dengan memasukkan nilai jenis warna HSV yang sudah ditentukan atau dengan sistem mencari yaitu mengetahui nilai jenis warna dari objek target sebagai masukan menggunakan sensor kinect, *mobile* robot dan pc harus terhubung. Untuk menghubungkan antara sensor kinect dan mikrokontroler yaitu arduino mega 2560 dengan pc menggunakan komunikasi serial. Sensor MQ-5 terhubung pin analog pada masukan mikrokontroler hasil data kandungan kadar gas yang dikirim secara komunikasi serial ke pc. Kendali *mobile* robot ini menggunakan jenis *driver* motor dc VN2SP30. *Mobile* robot ini di *supplay* dengan sumber baterai.

Ketika robot berjalan, data masukan dari nilai warna HSV yang sudah ditentukan atau dengan sistem mencari yang telah masuk, *mobile* robot akan *traking* mendekati objek. Dengan data *image* 640 pixel yang dibagi menjadi empat bagian berguna sebagai pengontrolan motor dc *mobile* robot bergerak dengan baik kemudian berhenti jarak antara *mobile* robot dengan objek yang sudah ditentukan dengan data luas dari objek. Informasi data error tersebut masuk ke dalam pemrosesan data, selanjutnya data tersebut digunakan sebagai acuan *mobile* robot.

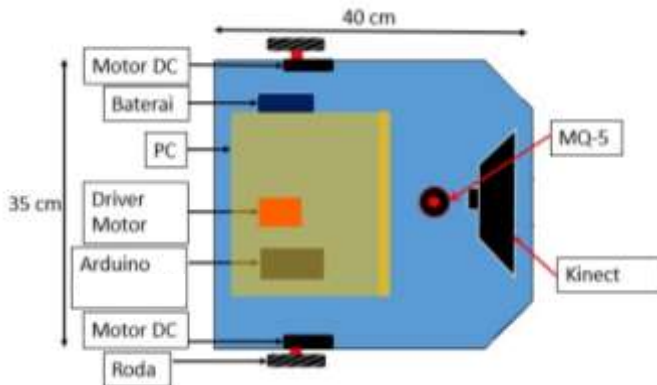
3.2 Desain Mekanik Robot

Desain mekanik *mobile* robot ini dimulai dengan perancangan perangkat keras kendali implementasi sensor kinect dengan penggabungan fungsi beberapa komponen seperti ditunjukkan Gambar 3.3 yaitu, sensor MQ-5, kinect v1, driver motor dc, motor dc 12volt, baterai, mikrokontroler dan pc. Semua komponen penyusun dirangkai sedemikian rupa seperti yang ditunjukkan pada Gambar 3.4 desain mekanik tampak atas dan Gambar 3.5 desain mekanik tampak atas sebagai desain *mobile* robot secara keseluruhan.

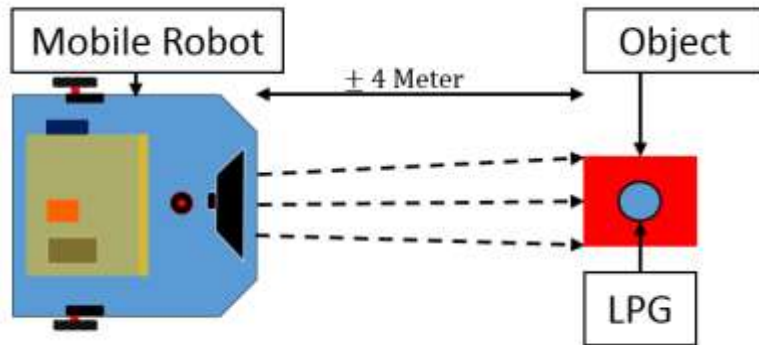
Pembuatan mekanik robot menggunakan desain yang dibuat oleh penulis. Model *mobile* robot yang digunakan adalah *mobile* robot bergerak dengan motor dc 2 buah dengan dengan masing-masing menggunakan satu roda belakang kiri dan kanan kemudian bagian depan menggunakan satu buah roda yang dapat bergerak ke semua arah dapat bergerak dengan baik.



Gambar 3. 3 Desain mekanik tampak samping



Gambar 3. 4 Desain mekanik tampak atas

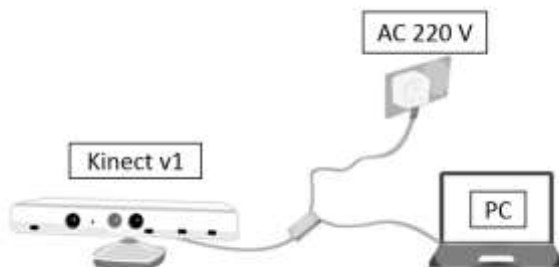


Gambar 3. 5 Algoritma *mobile robot*

Pada sistem algoritma *mobile robot* ini, pertama *mobile robot* telah mendeteksi objek warna, error data pixel dari sensor kinect mengirimkan ke mikrokontroler kemudian akan mengontrol motor dc dapat bergerak dengan baik menuju objek dan sensor MQ-5 mendeteksi kanduangan kadar gas.

3.2.1 Perancangan Sensor Kinect

Sensor kinect v1 merupakan komponen sensor yang menggunakan komunikasi serial yang dihubungkan usb sensor kinect pada pc. Pada jenis sensor kinect v1 sumber tegangan sensor yang masih terpisah dengan sumber tegangan pln ac 220 volt. Dapat dilihat pada Gambar 3.6 skema perancangan sensorr kinect v1.



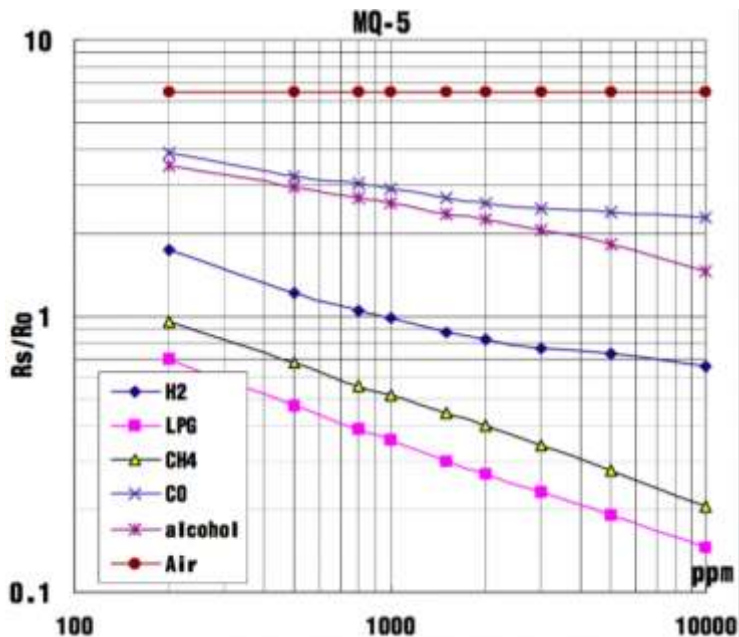
Gambar 3. 6 Perancangan sensor kinect

3.2.2 Perancangan Sensor MQ-5

Modul sensor MQ-5 mempunyai hasil keluaran berupa tegangan jika mendeteksi kadar gas *iso-butane* di dalam udara. Pin keluaran dari sensor gas ini dihubungkan dengan pin A0 pada arduino uno, sumber tegangan supply 5 volt dc dan *ground*. Sensor ini terdiri dari lapisan semikonduktor logam oksida yang terbentuk diatas sebuah substrat alumina pada sebuah sensing chip bersama dengan sebuah pemanas yang terintegrasi. Sensor ini bekerja berdasarkan perubahan resistansinya terhadap gas-gas tertentu termasuk asap. Tegangan untuk pemanas internal sangat penting. Beberapa sensor, menggunakan 5V untuk pemanas, yang lain perlu 2V. Beberapa langkah sensor untuk pemanasan ialah dengan menggunakan program `analogWrite ()` fungsi dan penundaan. Sebuah transistor atau logika tingkat MOSFET dapat digunakan untuk pemanas. Pemanas perlu selama sekitar 3 menit sebelum pembacaan menjadi stabil. Perancangan dapat dilihat pada Gambar 3.7.



Gambar 3. 7 Perancangan sensor MQ-5

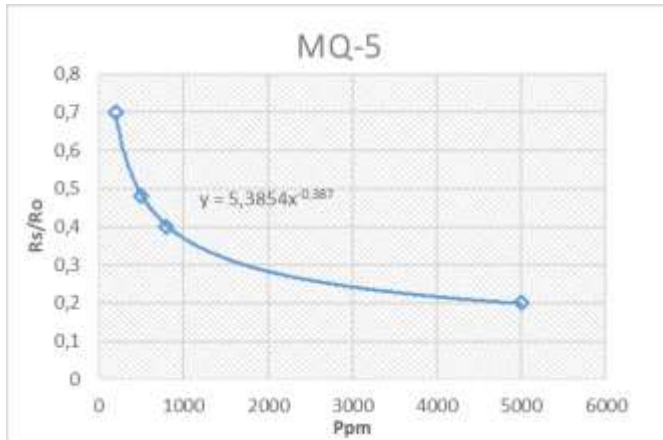


Gambar 3. 8 Grafik perbandingan Rs/Ro dengan kadar gas [4]

Dari Gambar 3.8 adalah grafik datasheet modul MQ-5 ppm (part per milion) terhadap RS/RO. Untuk mendapatkan nilai kadar gas, merujuka pada grafik sensor gas yang digunakan. Dalam penelitian ini menggunakan data grafik modul sensor MQ-5 pada nilai penginderaan gas *iso-butane*. Nilai dari grafik pada Gambar 3.8 kemudian diambil dan dimasukkan pada tabel. Sehingga mendapatkan tabel perbandingan antara Ro/Rs dengan ppm gas seperti yang ditunjukkan pada Tabel 3.1.

Tabel 3. 1 Nilai ppm gas terhadap perbandingan Ro/Rs

No.	Ro/Rs	Ppm
1	0,7	200
2	0,48	500
3	0,4	800
4	0,2	5000

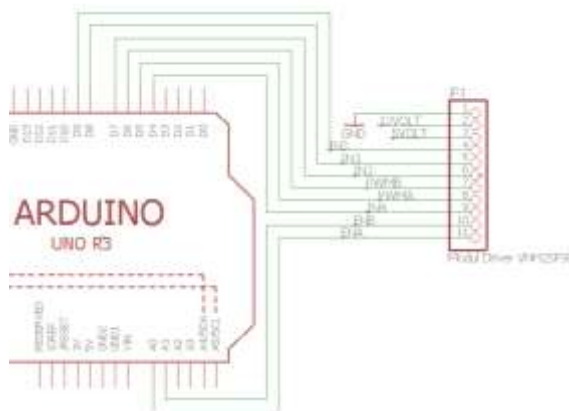


Gambar 3. 9 Grafik persamaan kadar ppm gas *iso-butane*

Nilai dari Tabel 3.1 kemudian di-plot pada grafik, untuk menghasilkan nilai kadar gas. Melalui persamaan yang dihasilkan oleh grafik pada Gambar 3.9.

2.9.1 Perancangan Driver Motor Dc VN2SP30

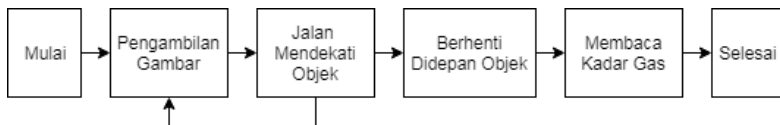
Modul driver motor dc VN2SP30 merupakan modul *H-Bridge* yang berbasis dua IC VN2SP30 *Half-Bridge* yang dikombinasikan menjadi satu. Konfigurasi dari rangkaian driver motor dc dengan arduino uno di tunjukan pada Gambar 3.10.



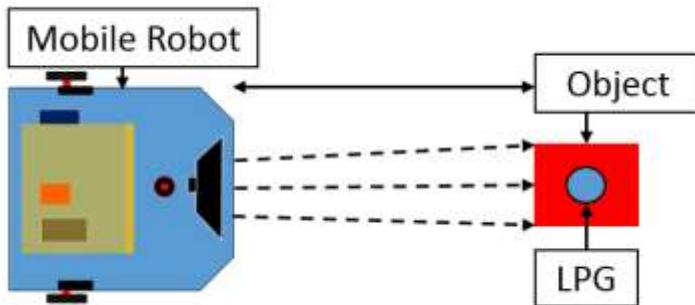
Gambar 3. 10 Perancangan driver motor dc VN2SP30

3.3 Perancangan Perangkat Lunak

Perancangan perangkat lunak ini digunakan untuk mengintegrasikan algoritma *mobile* robot dan memfungsikan modul-modul perangkat keras. Perancangan ini meliputi pengambilan gambar warna objek penunjukkan arah dari *mobile* robot yang dihasilkan oleh sensor kinect, jalan mendekati objek yang dihasilkan oleh kontrol driver motor dc, berhenti di depan objek yang dihasilkan sensor kinect dan membacakan kandungan kadar gas dihasilkan sensor MQ-5. Penjelasan ini yang tertera pada diagram blok algoritma *mobile* robot pada Gambar 3.11.



Gambar 3. 11 Diagram algoritma *mobile* robot



Gambar 3. 12 Perancangan diagram *mobile* robot



Gambar 3. 13 Diagram alur sistem *image*

Penentuan objek pada *mobile* robot menggunakan sensor kinect dapat dilihat pada Gambar 3.13 diagram alur sistem *image* yang implementasikan kamera dilakukan untuk mendapatkan pengambilan gambar objek, dilakukan mengubah RGB ke HSV, masukan warna dari objek HSV, setelah itu dilakukan filter mendapatkan data error yang baik yaitu dengan menggunakan filter erosi dan dilatasi. Error data pixel 640 dari sensor kinect diambil sumbu x yang kemudian di bagi menjadi dua data error yaitu positif 320 dan negatif 320 berfungsi sebagai penentuan titik tengah dari objek. Yang kemudian di kirim hasil data error dari pemrosesan citra ke pc.

3.3.1 Mengakses Sensor Kinect

Pada program dibawah ini menjelaskan jendela untuk menampilkan kedalaman dan gambar RGB. Seperti fitur lainnya yaitu apabila menekan 'w' pada keyboard menyebabkan motor dc pada sensor kinect ke atas, menekan 'x' menyebabkan kinect miring ke bawah dan apabila menekan '+' dan '-' menyebabkan berubahnya kontras warna. Ada beberapa opsi kontrol lain yang terdaftar di terminal ketika menjalankan *library* "freenect-glfw". Fungsi lain dari *library* ini adalah pengujian koneksi komunikasi serial dari sensor kinect terhadap pc.

```
void keyPressed(unsigned char key, int x, int y){
    if (key == 'w') {
        freenect_angle++;
        if (freenect_angle > 30){
            freenect_angle = 30;
        }
        tilt_changed++;
    }
    if (key == 'x'){
        freenect_angle--;
        if (freenect_angle < -30){
            freenect_angle = -30;
        }
        tilt_changed++;
    }
}
```

3.3.2 Mengakses HSV

```
VminH = cv2.getTrackbarPos('VminH', 'track')
VmaxH = cv2.getTrackbarPos('VmaxH', 'track')
VminS = cv2.getTrackbarPos('VminS', 'track')
VmaxS = cv2.getTrackbarPos('VmaxS', 'track')
VminV = cv2.getTrackbarPos('VminV', 'track')
VmaxV = cv2.getTrackbarPos('VmaxV', 'track')
```

Pada program di atas adalah mengakses HSV yang akan mendapatkan hasil fitur-fitur dibawah dengan menggunakan opencv dalam mengakses kamera kinect untuk menentukan hasil image processing dari metode-metode, seperti menampilkan mencari HSV dalam mentreking warna dari sebuah objek dengan mengubah-ubah min max hue, min max saturation dan min max value.

```
cv2.createTrackbar('VminH', 'track', 29, 180, nothing)
cv2.createTrackbar('VmaxH', 'track', 84, 180, nothing)
cv2.createTrackbar('VminS', 'track', 86, 255, nothing)
cv2.createTrackbar('VmaxS', 'track', 255, 255, nothing)
cv2.createTrackbar('VminV', 'track', 6, 255, nothing)
cv2.createTrackbar('VmaxV', 'track', 255, 255, nothing)
```

Pengambangan warna yang digunakan memisahkan warna hijau dengan yang warna lain. Pada ruang warna hsv, warna hijau memiliki nilai min hue 26 max hue 84, nilai min saturation 86 max 255 dan nilai value min 6 max 255. oleh karena itu, peng-ambangan warna dilakukan dengan batas hue mulai dari 0 sampai dengan 180, batas saturation mulai dari 0 sampai 255 dan valur mulai dari 0 sampai 255. Jangkauan nilai saturation dan value perlu diubah-ubah tergantung dari kondisi pencahayaan pada lapangan.

```
hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
```

Pada program diatas adalah pengubahan dari citra RGB ke citra HSV dengan hasil jendela penampil pada Gambar 3.15.

3.3.3 Mengakses Filter Erosi dan Dilatasi

```
frame = cv2.inRange(hsv, lower, upper)
frame = cv2.erode(frame, None, iterations=2)
frame = cv2.dilate(frame, None, iterations=2)
```

Pada program di atas setelah dilakukan segmentasi citra tahap selanjutnya adalah menghilangkan *noise*. Dikarenakan pada penelitian ini menggunakan sistem *image processing* maka kemungkinan adanya *noise* sangat mungkin terjadi. *Noise* dapat disebabkan karena kurangnya pencahayaan atau kondisi yang terlalu gelap. Selain itu juga dapat disebabkan oleh bayangan-bayangan air akibat air mengalami turbulensi. Oleh karena itu perlu dilakukan pem-*filter*-an untuk menghilangkan *noise* sebelum melakukan proses selanjutnya. *Filter* yang digunakan pada penelitian ini adalah *filter* erosi dan dilatasi.

3.3.4 Mengakses Pengukuran Jarak

```
if radius > 3:
    cv2.circle(img, (int(x), int(y)), int(radius), 0, 255, 255), 2)
    cv2.circle(img, center, 5, (0, 0, 255), -1)
    pts.appendleft(center)

if radius > 170:
    print "STOP"
    ser.write("777" + '\r\n')
```

Pada program di atas menunjukkan pengukuran jarak menggunakan sensor kinect dengan cara mengetahui nilai awal lingkaran dari ukuran pixel yang kemudian apabila ukuran lingkaran pixel semakin besar maka jarak antara sensor kinect dengan objek yang terdeteksi ukuran pixel semakin besar. Kemudian data error dikirim ke pc. Selanjutnya pc akan mengirim ke mikrokontroler untuk mengatur driver motor yang akan mengakibatkan roda kanan akan kirim berhenti.

3.3.5 Mengakses Sensor MQ-5

```
void loop() {
    float sensor_volt;
    float RS_air;
    float R0;
    float sensorValue;
    for(int x = 0 ; x < 100 ; x++)
    {sensorValue = sensorValue + analogRead(A0);}
    sensorValue = sensorValue/100.0;
    sensor_volt = sensorValue/1024*5.0;
    RS_air = (5.0-sensor_volt)/sensor_volt;
    R0 = RS_air/0.7;
```

Pada program diatas adalah untuk mencari nilai ro mendapatkan ro/rs. Pengukuran disini mengambil jenis LPG yang tinggi respon terhadap sensor MQ-5. Dengan nilai ro/rs terhadap ppm dari grafik datasheet yang tertera pada Gambar 3.10 adalah 0.7. sensor MQ-5 ini sangat berpengaruh terhadap tekanan udara dan suhu. Nilai awal untuk mengetahuinya dengan melihat nilai output analog adc dari sensor MQ-5 yang kemudian dibagi 5volt masukan tegangan supply.


```

void setup(){
  Serial.begin(9600);}
void loop(){
  float ppm;
  float sensor_volt;
  float RS_gas;
  float ratio;
  int sensorValue = analogRead(A1);
  sensor_volt=(float)sensorValue/1024*5.0;
  RS_gas = (5.0-sensor_volt)/sensor_volt;
  ratio = RS_gas/19.98;
  ppm = 77.206*pow(ratio,-2.58);
  Serial.print(sensor_volt);
  Serial.print(RS_gas);
  Serial.print(ratio);
  Serial.print(sensorValue);
  Serial.print(ppm);
  Serial.print(millis());
  delay(1000);}

```

Pada program diatas masukan nilai R0 yang telah di dapat untuk menghasilkan nilai kandungan kadar gas *iso-butane* kemudian akan mendapatkan nilai ro/rs dan di ubah menjadi nilai ppm. Dalam perancangan ini ro/rs terhadap waktu sama halnya juga ppm.

-----Halaman ini sengaja dikosongkan-----

BAB IV

PENGUJIAN DAN PEMBAHASAN SISTEM

Pada bab ini, dibahas mengenai pengujian dari sistem yang telah dikembangkan. Bab ini bertujuan untuk mengetahui apakah permasalahan dalam perancangan sistem pada tugas akhir ini telah terselesaikan atau tidak. Pengujian pada bab ini terdiri dari pengujian penentuan objek dengan sensor kinect, pengujian mendapatkan nilai warna hsv, pengujian pengukuran jarak dengan sensor kinect, pengujian sensor MQ-5 dan pengujian sistem *mobile* robot.

4.1 Hasil Realisasi Desain Mekanik *Mobile* Robot

Desain mekanik *mobile* robot ini dimulai dengan perancangan perangkat keras kendali implementasi sensor kinect dengan penggabungan fungsi beberapa komponen seperti ditunjukkan Gambar 3.4 yaitu, sensor MQ-5, kinect v1, driver motor dc, motor dc 12volt, baterai, mikrokontroler dan pc. Semua komponen penyusun dirangkai sedemikian rupa seperti yang ditunjukkan pada Gambar 4.1 desain mekanik tampak depan

Pembuatan mekanik robot menggunakan desain yang dibuat oleh penulis. Model *mobile* robot yang digunakan adalah *mobile* robot bergerak dengan motor dc 2 buah dengan dengan masing-masing menggunakan satu roda belakang kiri dan kanan kemudian bagian depan menggunakan satu buah roda yang dapat bergerak ke semua arah dapat bergerak dengan baik.



Gambar 4. 1 Realisai desain sistem *mobile* robot

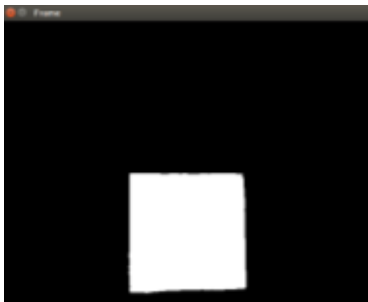
4.2 Pengujian Pengolahan Citra



Gambar 4. 2 Citra asli



Gambar 4. 3 Citra hasil perubahan RGB ke HSV



Gambar 4. 4 Citra hasil filter dilatasi dan erosi



Gambar 4.5 Citra hasil proses perolehan fitur

Pengujian dilakukan dengan mengatur batas-batas nilai ambang yang digunakan untuk memperoleh citra biner yang memiliki warna pada citra RGB warna hijau dan warna hitam pada RGB warna selain hijau. Nilai batas ambang ini perlu diatur pada setiap lingkungan yang berbeda. Pada Gambar 4.2 menunjukkan citra asli, Gambar 4.3 citra yang menunjukkan hasil perubahan dari citra RGB ke HSV, Gambar 4.4 citra hasil filter erosi dan dilatasi, pada Gambar 4.5 Citra hasil proses perolehan fitur akhir yang ditambah dengan menampilkan nilai dari pembacaan adc sensor MQ-5.

4.3 Pengujian Mendapatkan Nilai Warna HSV



Gambar 4. 6 Pengujian mencari masukan nilai HSV



Gambar 4. 7 Pengujian hasil mencari manual berwarna hijau dengan indikator percahayaan 2000 lux



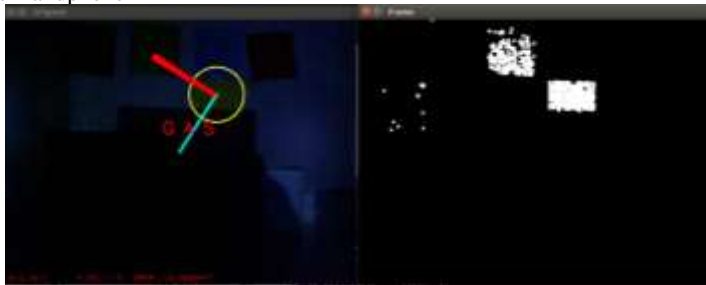
Gambar 4. 8 Hasil pengujian percahayaan lux meter dengan menggunakan aplikasi handphone



Gambar 4. 9 Pengujian hasil mencari manual berwarna hijau dengan indikator percahayaan 225 lux



Gambar 4. 10 Hasil pengujian percahayaan lux meter dengan menggunakan aplikasi handphone



Gambar 4. 11 Pengujian hasil mencari manual berwarna hijau berubah deteksi menjadi kuning akibat percahayaan 80 lux



Gambar 4. 12 Hasil pengujian percahayaan lux meter dengan menggunakan aplikasi handphone

Pada pengujian ini dilakukan untuk pengambilan nilai warna yang dicari di dapat terdeteksi dengan cara merubah-ubah nilai range hue, saturation dan value dengan hasil yang dapat dilihat pada Tabel 4.1. Contoh dalam pengambilan nilai mencari dapat dilihat di Gambar 4.6 yaitu mencari warna hijau pada Gambar 4.7 pada dengan nilai HSV adalah hue min 29 max 84, saturation min 86 max 255 dan value min 6 max 255. Dengan percahayaan 2000 lux pada Gambar 4.8. pada Gambar 4.9 perubahan percahayaan sangat berpengaruh terhadap pengujian ini. Pada pengujian ini percahayaan yang ideal dan ditempat dalam ruangan. Gambar 4.10 pada saat pengujian hasil mencari manual berwarna hijau yang berubah deteksi warna menjadi kuning pada Gambar 4.11 akibat percahayaan 80 lux pada Gambar 4.12.

Tabel 4. 1 Hasil mencari nilai color HSV

No.	Color	Hmin	Hmax	Smin	Smax	Vmin	Vmax
1	Biru muda	102	110	86	255	6	255
2	Biru tua	102	122	86	255	6	255
	Hijau tua	47	109	0	255	0	176
3	Hijau muda	24	50	86	255	6	255
4	Orangen tua	0	13	86	255	7	255
5	Orangen muda	8	28	136	255	0	255
6	Kuning muda	19	28	136	255	0	255
7	Kuning tua	17	34	115	255	10	216
8	Merah muda	167	180	198	245	112	148
9	Merah tua	171	180	158	255	0	255
12	Coklat	0	74	54	134	110	126

4.4 Pengujian Pengukuran Jarak Menggunakan Depth Sensor Kinect

Pada pengujian jarak menggunakan Depth sensor kinect dapat dilihat pada Gambar 4.13 yaitu implementasi pengujian sensor depth pada objek yang berupa semua objek benda. Pengujian ini dilakukan untuk mengetahui jarak minimal dan maksimal depth sensor kinect dapat pendeteksi objek benda di depannya. Hasil dari pengukuran dari depth sensor kinect kemudian dibandingkan dengan melakukan pengukuran manual dengan menggunakan alat ukur penggaris. Dapat dilihat hasil data dari depth sensor kinect dengan objek benda yang ada didepannya pada hasil Gambar 4. 14. Jarak perubahan diambil setiap 10 cm terhadap setiap langkah hasil pembacaan.

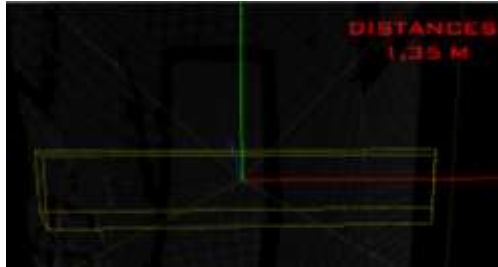
Dari hasil yang didapat pada Gambar 4.14 mengetahui jarak minimal depth sensor kinect bekerja adalah 40 cm kemudian jarak maksimal depth sensor kinect mengetahui jarak adalah 400 cm. Dan hasil antara pengukuran manual dengan pengukuran menggunakan depth sensor kinect presisi dan akurat.



Gambar 4. 13 Implementasi pengujian sensor depth pada objek



Gambar 4. 14 Grafik pengujian jarak menggunakan depth sensor kinect



Gambar 4. 15 Interface depth sensor

Pada Gambar 4.15 adalah interface depth sensor hasil pengujian tampilan dari pembacaan depth sensor kinect dalam mengetahui jarak membacaan antara sensor kinect terhadap objek yang ada di depannya.

4.5 Pengujian Pengukuran Jarak Objek Dengan Luasan Objek



Gambar 4. 16 Jarak sensor kinect pada *mobile* robot dengan objek

Tabel 4. 2 Hasil data pengukuran jarak Objek dengan *mobile* robot

No.	Masukan luasan resolusi kinect (pixel)	Jarak terukur manual(cm)	Resolusi kinect sumbu x (pixel)
1	270	9	640
2	250	12	640
3	230	15	640
4	210	17	640
5	190	21	640
6	170	24	640
7	150	35	640
8	130	48	640
9	110	58	640
10	90	73	640

Pada pengujian ini masukan luasan resolusi kinect berfungsi sebagai berhentinya *mobile* robot terhadap objek target. Kemudian

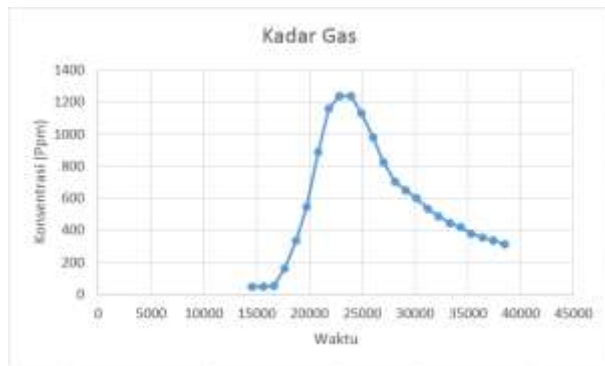
melakukan pengukuran manual setiap langkah masukan luasan. Pada pengujian didapat hasil pengukuran dengan jarak berhenti *mobile* robot yang dapat diatur dapat dilihat pada Tabel 4.2 hasil data pengukuran jarak *mobile* robot dengan objek menggunakan sensor kinect, pixel dari kamera 640 dengan menggunakan sumbu x. Masukan luas lingkaran sebagai perbandingan untuk berhentinya *mobile* robot. Kemudian melakukan pengukuran manual dengan menggunakan penggaris. Maka akan terlihat ketika masukan luas lingkaran semakin kecil yaitu 90 pixel, maka jarak pengukuran manual akan semakin besar yaitu 73 cm. Pada pengujian jarak maksimal adalah 22 meter dengan percahayaan 2000 lux dan jarak minimal 9 cm objek terhadap sensor kinect dapat dilihat pada Gambar 4.16.

4.6 Pengujian Sensor MQ-5

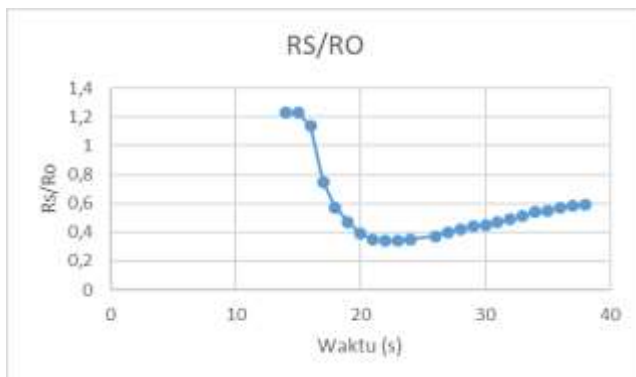
Pada tahap ini pengujian sensor MQ-5 berupa gas *iso-butane*. Kalibrasi yang dilakukan tanpa adanya zat *iso-butane* pada area tempat sensor tersebut bekerja. Kalibrasi sensor dilakukan dengan menempatkan kadar gas ke sensor MQ-5 pada area tempat kerja sensor tersebut. Sensor diletakan pada area dengan kadar udara netral dari gas *iso-butane* untuk mendapatkan nilai Ro. Dilakukan di dalam ruangan dengan suhu 25°C, kecepatan kipas *Air Conditioner* pada skala 3 (Panasonic Eco Smart ruang 402). Data hasil pengujian respon sensor kemudian berupa perubahan perbandingan R_s/R_o , Ppm dan perubahan konsentrasi gas terhadap waktu t (s) dapat dilihat pada Gambar 4.18 dan 4.19 Dengan jarak sumber tabung gas *iso-butane* terhadap sensor MQ-5 adalah 1 meter. Dapat dilihat pada tabel 4.3 hasil nilai pengujian sensor MQ-5 terhadap gas *iso-butane*. Pengujian ini berpengaruh terhadap waktu dapat terlihat ketika dibuat grafik ketika sensor MQ-5 mendeteksi gas *iso-butane*. Dan pada Gambar grafik 4.19, dari waktu yang berpengaruh terhadap kadar gas dari pembacaan nilai R_s/R_o . Ppm merupakan kadar gas yang terdeteksi, t merupakan waktu dalam satuan detik. Dapat dilihat dari grafik Gambar 4.18, dari t saat 14 s hingga t saat 38 s, pembacaan gas stabil ketika adanya kadar *iso-butane* pada hasil data Tabel 4.3 bahwa nilai Ppm 1129,27 dan ketika tidak adanya kadar gas *iso-butane* nilai Ppm 45,14 maka sesuai dengan Gambar 3.9 grafik dari karakteristik sensor MQ-5 dapat disimpulkan bahwa ketika adanya kadar gas *iso-butane* maka nilai Ppm semakin besar.



Gambar 4. 17 Pengujian *mobile robot* dengan gas *iso-butane*



Gambar 4. 18 Grafik hasil data sensor gas MQ-5 ppm terhadap waktu



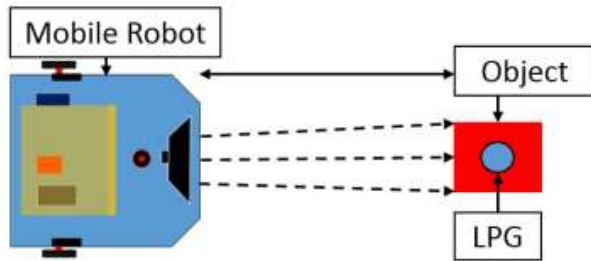
Gambar 4. 19 Grafik hasil data sensor gas MQ-5 Rs/Ro terhadap waktu

Tabel 4. 3 Hasil data pemberian gas iso-butane

No.	Vcc (Volt)	Masukan A0 (Volt)	RS-ratio (Volt)	Ppm	Sensor Value	RS/RO	Sekon
1	5	0,2	24,6	45,14	40	1,23	14
2	5	0,2	24,6	45,14	43	1,23	15
3	5	0,21	22,81	54,83	64	1,14	16
4	5	0,31	15	161,76	83	0,75	17
5	5	0,41	11,34	333,08	99	0,57	18
6	5	0,48	9,34	548,63	117	0,47	19
7	5	0,57	7,75	888,14	128	0,39	20
8	5	0,63	7	1155,67	131	0,35	21
9	5	0,64	6,82	1237,51	131	0,34	22
10	5	0,64	6,82	1237,51	127	0,34	23
11	5	0,62	7,06	1129,27	121	0,35	24
12	5	0,59	7,46	979,71	114	0,37	26
13	5	0,56	7,98	823,52	108	0,4	27
14	5	0,53	8,48	704,26	105	0,42	28
15	5	0,51	8,75	649,38	102	0,44	29
16	5	0,5	9,04	597,54	98	0,45	30
17	5	0,48	9,45	532,96	95	0,47	31
18	5	0,46	9,78	487,79	92	0,49	32
19	5	0,45	10,13	445,31	90	0,51	33
20	5	0,44	10,77	418,44	87	0,54	34
21	5	0,42	11,05	380,24	85	0,55	35
22	5	0,42	11,34	356,13	83	0,57	36
23	5	0,41	11,64	333,08	81	0,58	37
24	5	0,4	11,8	311,06	80	0,59	38

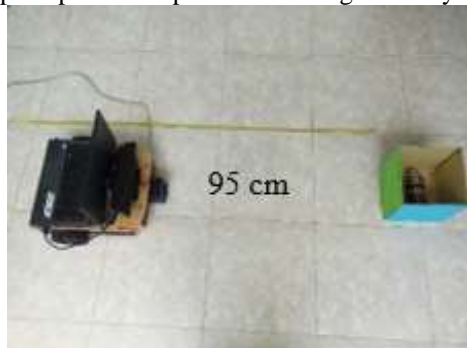
4.7 Pengujian Integrasi Sistem *Mobile Robot*

4.7.1 Posisi Objek di Depan *Mobile Robot*

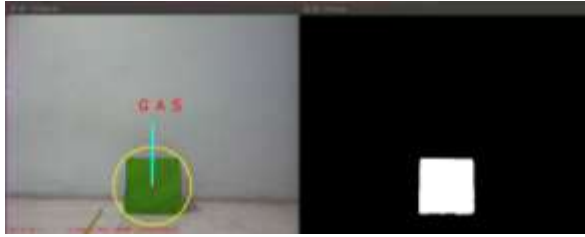


Gambar 4. 20 Algoritma sistem *mobile robot* pada posisi objek di depan

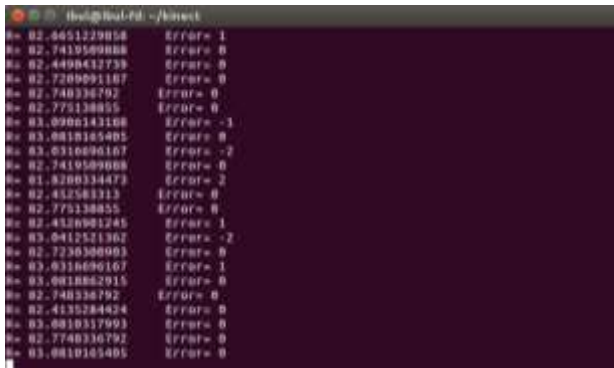
Pada pengujian ini adalah pengujian yang dilakukan pada sistem yang sudah terintegrasi untuk kemudian diuji pada ruangan lapangan tertutup. Pengujian ini dilakukan ruangan kost dengan panjang 4 x 8. Dengan pengujian pertama yaitu Gambar 4.21 simulasi yang akan dijadikan algoritma posisi objek didepan *mobile robot* dan Gambar 4.22 hasil implementasi dari *mobile robot* sebenarnya dengan objek yang berwarna hijau. Dengan jarak 95 cm dari objek terlihat pada Gambar 4.22 dengan alat ukur meteran kain. Gambar 4.20 menunjukkan citra dari hasil filter dilatasi dan erosi. kemudian pada Gambar 4.23 citra hasil proses perolehan fitur tampak depan dengan sumbu x ditengah dengan data error kurang lebih 0 artinya hasil implementasi sensor kinect dengan objek pada posisi didepan terbaca dengan hasil yang tepat.



Gambar 4. 21 Hasil implementasi algoritma sistem *mobile robot* pada posisi objek di depan

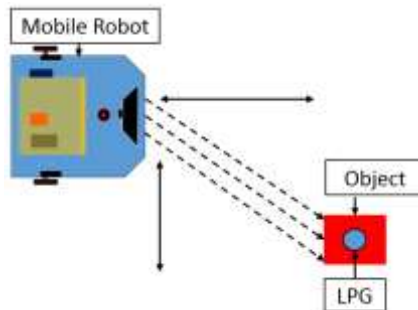


Gambar 4. 22 Citra hasil filter dilatasi dan erosi dan citra hasil proses perolehan fitur tampak depan



Gambar 4. 23 Hasil data error dari deteksi objek tampak depan

4.7.2 Posisi Objek di Kanan *Mobile Robot*

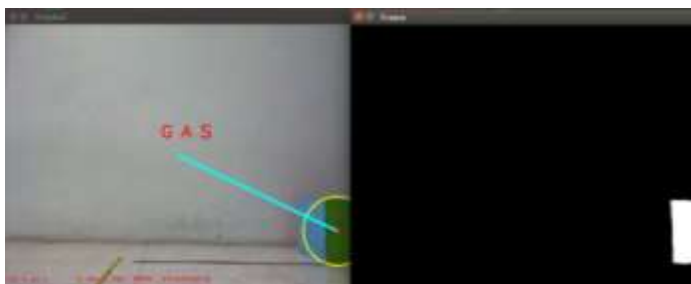


Gambar 4. 24 Algoritma sistem *mobile robot* pada posisi objek di kanan

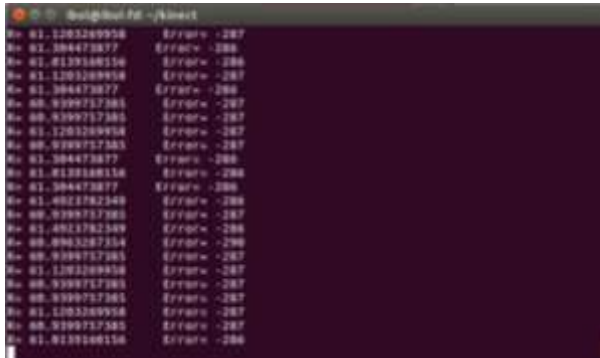


Gambar 4. 25 Hasil implementasi algoritma sistem *mobile* robot pada posisi objek di kanan

Pada pengujian yang kedua yaitu dengan melakukan simulasi dan implementasi. Pengujian ini dengan melakukan posisi objek berwarna hijau di sebelah kanan pada *mobile* robot dengan jarak depan 95 cm dan jarak samping 60 cm pada Gambar 4.25. Simulasi dan implementasi ini pada Gambar 4.24 dilakukan untuk mengetahui maksimal panjang sumbu x pada sensor kinect dalam masukan pixel 640 yang di bagian kanan yaitu dengan nilai error data 0 sampai -286 pada Gambar 4.27. Pada Gambar 4.26 Citra hasil filter dilatasi dan erosi dan citra hasil proses perolehan fitur tampak kanan. Data error pembacaan objek dari sensor kinect ini yang kemudian akan dikirim dari pc ke mikrokontroler dengan komunikasi serial, kemudian mikrokontroler akan mengontrol driver motor dc *mobile* robot dapat berjalan sesuai dengan data yang diterima.



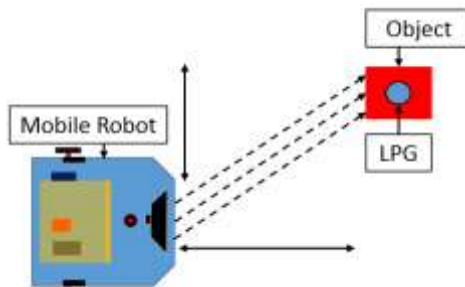
Gambar 4. 26 Citra hasil filter dilatasi dan erosi dan citra hasil proses perolehan fitur tampak kanan



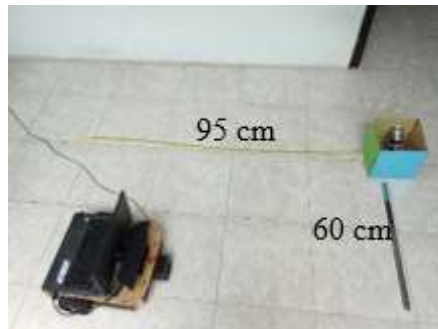
Gambar 4. 27 Hasil data error dari deteksi objek tampak kanan

4.7.3 Posisi Objek di Kiri *Mobile Robot*

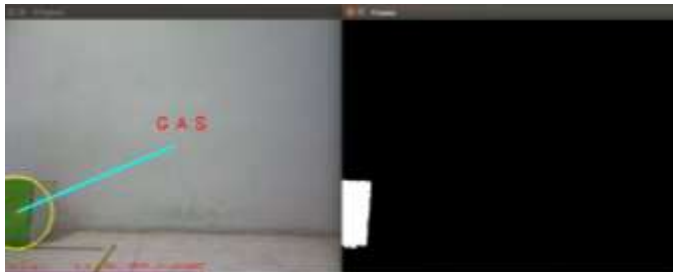
Pada pengujian yang kedua yaitu dengan melakukan simulasi dan implementasi pada Gambar 4.28. Pengujian ini dengan melakukan posisi objek berwarna hijau pada Gambar 4.30 dengan filter dilatasi dan erosi di sebelah kiri pada *mobile robot* dengan jarak depan 95 cm dan jarak samping 60 cm pada Gambar 4.29. Simulasi dan implementasi ini dilakukan untuk mengetahui maksimal panjang sumbu x pada sensor kinect dalam masukan pixel 640 yang di bagian kiri yaitu sumbu x dengan nilai error data 0 sampai 284 pada Gambar 4.31. Berarti titik tengah pada sumbu x ini adalah 0 terlihat pada Gambar 4.29. Data error pembacaan objek dari sensor kinect ini yang kemudian akan dikirim dari pc ke mikrokontroler dengan komunikasi serial, kemudian mikrokontroler akan mengontrol driver motor dc *mobile robot* dapat berjalan sesuai dengan data yang diterima.



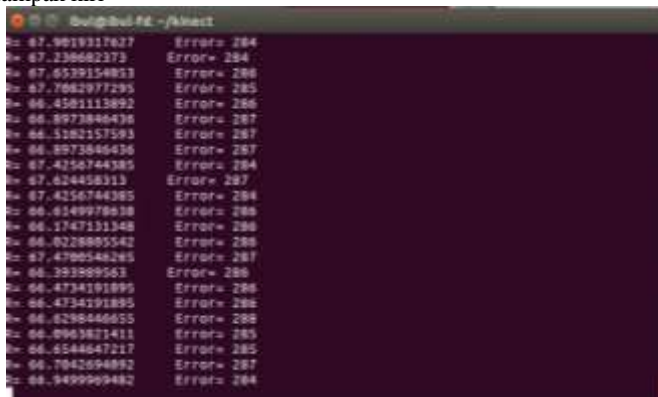
Gambar 4. 28 Algoritma sistem *mobile robot* pada posisi objek di kiri



Gambar 4. 29 Hasil implementasi algoritma sistem *mobile robot* pada posisi objek di kiri



Gambar 4. 30 Citra hasil filter dilatasi dan erosi dan citra hasil proses perolehan fitur tampak kiri



Gambar 4. 31 Hasil data error dari deteksi objek tampak kiri

-----Halaman ini sengaja dikosongkan-----

BAB V

PENUTUP

1.1 Kesimpulan

Berdasarkan data hasil uji implementasi *mobile* robot ini diperoleh beberapa kesimpulan antara lain dua komponen yang penting dalam sistem *mobile* robot ini yaitu sensor kinect yang sangat berpengaruh terhadap membacaan objek yang akan dijadikan target dan berjalannya *mobile* robot. Hasil percobaan kamera sensor kinect ketika mendapatkan nilai objek warna hijau dengan metode HSV adalah *hue min 29 max 84, saturation min 86 max 255, value min 6 max 255* dengan percahayaan ruangan 2000 lux, kesalahan nilai warna objek hijau dengan percahayaan ruangan 80 lux. Ketika percahayaan 80 lux mengakibatkan sensor kinect tidak dapat mengenali atau tidak dapat mendekteksi warna objek target. Ketidakakuratan dari sensor kinect saat mengenali atau mendektisi objek target maka arah berjalan *mobile* robot tidak sesuai dengan yang diinginkan.

Pada pengukuran jarak menggunakan depth sensor kinect yang dilakukan, didapat bahwa minimal kinect adalah 40 cm dalam membaca objek dan maksimanl 400 cm. Apabila menggunakan pengujian program inputan luasan objek dengan pengukuran jarak objek dengan kamera kinect adalah jarak minimum terukur 9 cm dan jarak maksimum 22 meter dengan percahayaan 2000 lux.

Kecepatan dan arah angin dapat mempengaruhi nilai pembacaan sensor gas MQ-5 ini. pembacaan gas stabil ketika adanya kadar *iso-butane* pada hasil data Tabel 4.3 bahwa nilai Ppm 1129,27 dan ketika tidak adanya kadar gas *iso-butane* nilai Ppm 45,14. maka sesuai dengan Gambar 3.9 grafik dari karakteristik sensor MQ-5 dapat disimpulkan bawah ketika adanya kadar gas *iso-butane* maka nilai Ppm semakin besar. Pada sensor gas dapat memetakan kadar gas suatu lokasi kurang lebih 1 meter dalam keadaan ruang hampa.

Pada pengujian hasil algoritma integrasi sistem *mobile* robot ini yang dilengkapi dengan sensor kinect. Dengan objek target warna yang masih terlihat oleh sensor kamera kinect dan lokasi area yang datar *mobile* robot dapat berjalan. Error data deteksi objek adalah 284 pixel untuk bagian kiri dan bagian kanan error data -286 pixel, error data ini sebagai acuan *mobile* robot berjalan menuju objek target.

1.2 Saran

Penambahan sensor jarak yang berfungsi sebagai mendeteksi halangan *mobile* robot. Pemasangan GPS yang berfungsi sebagai *autonomous mobile* robot bergerak lebih jauh jangkauan dan dapat navigasi otomatis. Penambahan jumlah sensor gas pada empat sisi robot, untuk pendeteksi gas dari segala arah. Dan penggunaan processor yang lebih ideal pada *mobile* robot seperti mini pc intel.

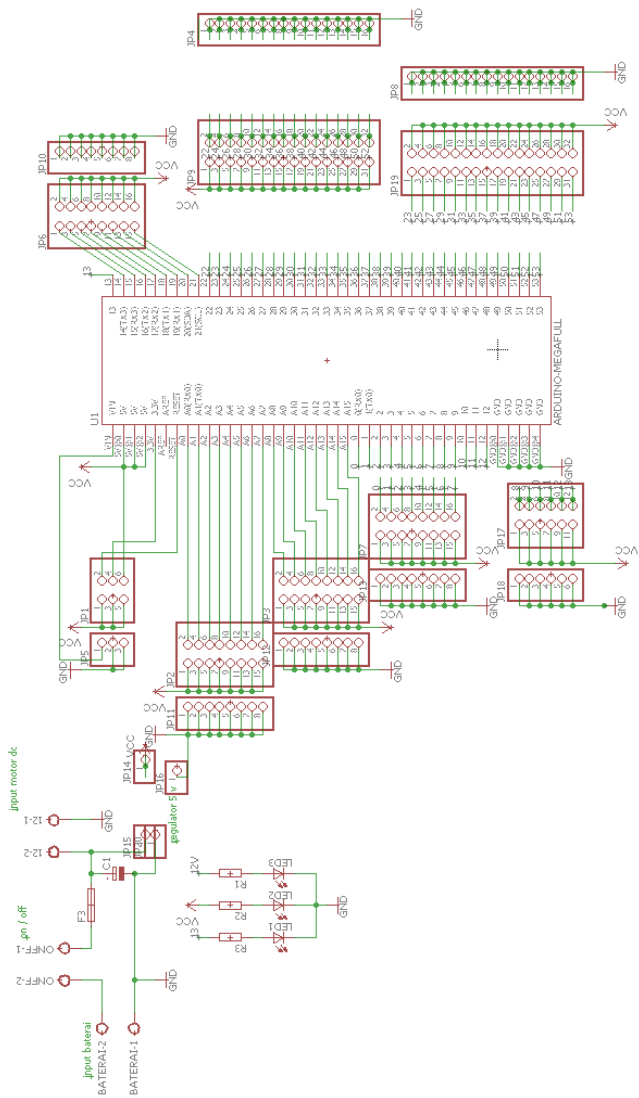
DAFTAR PUSTAKA

- [1] S. Riyanta, “Prediksi dan Analisis Ancaman Terorisme Tahun 2017 di Indonesia,” Detiknews: <https://news.detik.com/kolom/d-3387780/prediksi-dan-analisis-ancaman-terorisme-tahun-2017-di-indonesia?single=1>. Diakses: 09-Jun-2018.
- [2] Niu Zhigang, Wang Lu, “Hazardous Gas Detecting Method Applied in Coal Mine Detection Robot,” Cina, University of Technology, Taiyuan, Shanxi, 2011.
- [3] N. Zhigang dan W. Lu, “Hazardous Gas Detecting Method Applied in Coal Mine Detection Robot,” dalam *2011 Third International Conference on Measuring Technology and Mechatronics Automation*, 2011, vol. 2, hlm. 308–311.
- [4] Roby Adi Wibobo, Muhammad Rivai, dan Suwito, “Implementasi Autonomous Navigation Robot Menggunakan Global Positioning System (Gps) Untuk Pemetaan Kadar Gas Berbahaya,” Surabaya: ITS, Jan 2017.
- [5] Datasheet Sensor MQ-5: <https://www.parallax.com/sites/default/files/downloads/605-00009-MQ-5-Datasheet.pdf>. Diakses: 04-Jun-2018.
- [6] N. A. J. R. A. N. João Silva, Cooperative detection and identification of obstacles in a robotic, dalam *Advances in Robotics: Modeling, Control and Applications*, Portugal, iConcept Press, 2014, pp. 3 - 5.
- [7] Adnan Rachmanwan, Hendra Kusuma, dan Ronny Mardiyanto, “Penentuan Posisi Robot Sepak Bola Beroda Menggunakan Rotary Encoder Dan Kamera,” Surabaya: ITS, Agu 2017.
- [8] Muhammad Yunus Andrian, Djoko Purwanto, Ronny Mardianto, “Penerjemahan Bahasa Isyarat Indonesia Menggunakan Kamera pada Telepon Genggam Android,” Surabaya: ITS, 2016.
- [9] R. E. W. Rafael C. Gonzalez, *Digital Image Processing Second Edition*, New Jersey : Prentice-Hall, Inc., 2002.
- [10] Tjokorda Agung Budi Wirayuda, “Pemanfaatan Operasi Morphologi Untuk Proses Pendeteksian Sisi Pada Pengolahan Citra Digital,” hlm. 8, Bandung: Telkom: Nov 2006.
- [11] Tjokorda Agung Budi Wirayuda, ST, “Pemanfaatan Operasi Morphologi Untuk Proses Pendeteksian Sisi Pada Pengolahan Citra Digital,” Bandung: Telkom, 2006.

- [12] Yusril Izza dan S. T. Budi Bayu Murti, “Sistem Kontrol Gerak Lengan Robot Menggunakan Metode Isyarat Gerak Tubuh Dengan Sensor Kinect,” Yogyakarta: Universitas Gadjah Mada, 2016.
- [13] OpenCV library : <https://opencv.org/>. Diakses: 12-Jun-2018.
- [14] Rangga Aditya S, Muhammad Arrofiq, “Pengenal Plat Nomor Kendaraan Berbasis Raspberry Pi,” Yogyakarta: Universitas Gadjah Mada, 2016.
- [15] M. Syaif Ramadhan dan Muhammad Rivai, “Sistem Kontrol Tingkat Kekeruhan pada Aquarium Menggunakan Arduino Uno,” J. Tek. ITS, vol. 7, Mar 2018.
- [16] *Datasheet* Arduino Uno Rev3: <https://store.arduino.cc/usa/arduino-uno-rev3>. Diakses: 12-Jun-2018.
- [17] Ogata, katsuhiko, Edi Laksono (Penterjemah). 1993. Teknik Kontrol Otomatik (Sistem Pengaturan) Jilid 2. Jakarta : Erlangga.
- [18] myboticFollow, “Tutorial for Monster Motor Shield VNH2SP30,” *Instructables.com*: <http://www.instructables.com/id/Monster-Motor-Shield-VNH2SP30/>. Diakses: 12-Jun-2018.
- [19] *Datasheet* : <https://www.st.com/en/automotive-analog-and-power/vnh2sp30-e.html>. Diakses: 12-Jun-2018.
- [20] Spoiled Meat Classification Using Semiconductor Gas Sensors, Image Processing and Neural Network,” Surabaya: ITS, Mar 2018.

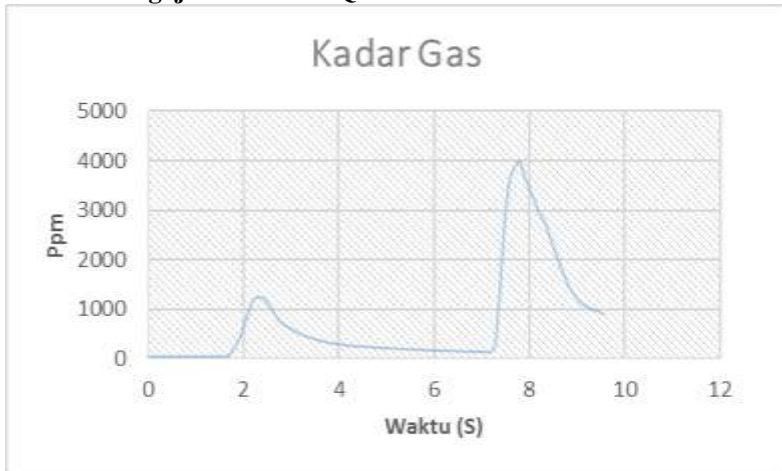
LAMPIRAN

A. Desain Elektronis *Mobile Robot*

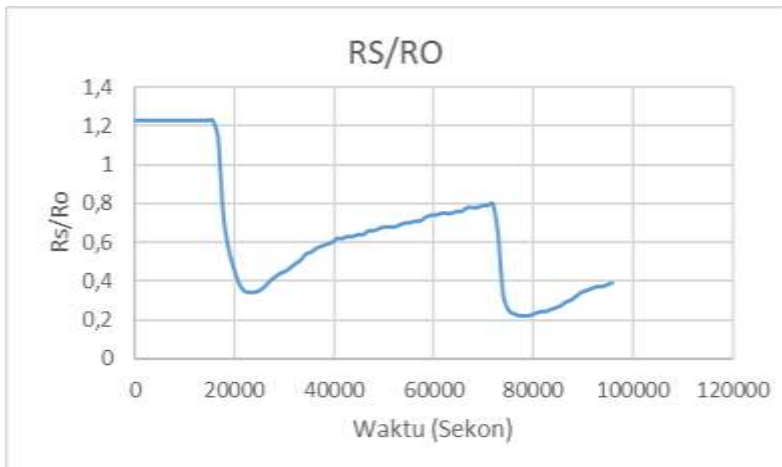


Gambar Desain Skematik

B. Data Pengujian Sensor MQ-5



Gambar Grafik Sensor Gas MQ-5 Waktu Terhadap PPM



Gambar Grafik Sensor Gas MQ-5 Waktu Terhadap Rs/Ro

C. Program Arduino

```
#define enA A0
#define pwmA 5
#define in1 8
#define in2 7
#define enB A1
#define pwmB 6
#define in3 9
#define in4 4

float sensor = A5;
float gas_value;
String inputString = "";
unsigned int rat, pul, sat;
String dataIn;
String dt[10];
int i;
boolean parsing=false;
    int data=0 ;
    int luas=0 ;
// defines variables
long duration;
int distance;

void maju ()
{
    digitalWrite (in1, LOW);
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
}
void kanan()
{
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
}

void kiri ()
{
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
}
void putarkanan()
{
    digitalWrite(in1, LOW);
```

```

    digitalWrite(in2, HIGH);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    analogWrite(pwmA, 50);
    analogWrite(pwmB, 50);
}

void putarkiri()
{
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    analogWrite(pwmA, 50);
    analogWrite(pwmB, 50);
}

void HOP()
{
    gas_value = int (analogRead(sensor));
    Serial.print(int(gas_value / 100));
    Serial.print(int((int(gas_value) % 100) / 10));
    Serial.print(int((int(gas_value) % 100) % 10));
    delay(200);

    pinMode(enA, HIGH);
    pinMode(enB, HIGH);

    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    analogWrite(pwmA, 0);
    analogWrite(pwmB, 0);
}

void setup() {
    Serial.begin(9600); // Starts the serial communication
    dataIn="";
    pinMode(pwmA, OUTPUT);
    pinMode(enA, OUTPUT);
    pinMode(in1, OUTPUT);
    pinMode(in2, OUTPUT);

    pinMode(pwmB, OUTPUT);
    pinMode(enB, OUTPUT);
    pinMode(in3, OUTPUT);
    pinMode(in4, OUTPUT);

    pinMode(enA, LOW);
    pinMode(enB, LOW);
}

```

```

    pinMode(sensor, INPUT);
    Serial.begin(9600);
}
void loop()
{
    track();
}
void track()
{
    int flag = 0;
    if (Serial.available()) {
        char inChar = (char)Serial.read();
        dataIn += inChar;
        if (inChar == '\n') {
            parsing = true;
        }
    }

    if(parsing){
        parsingData();
        parsing=false;
        dataIn="";
    }
    int KP = 4;
    float P = KP * (luas / 10);
    if ( luas < 75)
    {
        P = 0;
    }
    if ( luas > 170)
    {
        HOP();
    }
    else if ( data == 999)
    {
        if (flag == 1)
        {
            putarkanan();
        }
        else if (flag == 2)
        {
            putarkiri();
        }
    }
    else if (data >= 200)
    {
        //kanan
        kanan();
        flag = 1;
        analogWrite(pwmA, 125 - P);
    }
}

```

```

        analogWrite(pwmB, 125 - P);
    }
    else if (data >= 100 && data < 200 ) //100-200
    {
        flag = 1;
        maju();
        analogWrite(pwmA, 100 - P);
        analogWrite(pwmB, 75 - P);
    }
    else if (data >= 50 && data < 100 ) //50 - 100
    {
        maju();
        flag = 1;
        analogWrite(pwmA, 100 - P);
        analogWrite(pwmB, 85 - P);
    }
    else if (data > -50 && data < 50 ) //-50 - 50
    {
        maju();
        flag = 0;
        analogWrite(pwmA, 85 - P);
        analogWrite(pwmB, 85 - P);
    }
    else if (data >= -100 && data < -50 ) //-100 - 50
    {
        flag = 2;
        maju();
        analogWrite(pwmA, 85 - P);
        analogWrite(pwmB, 100 - P);
    }
    else if (data >= -200 && data < 100 ) //-200-100
    {
        flag = 2;
        maju();
        analogWrite(pwmA, 75 - P);
        analogWrite(pwmB, 100 - P);
    }
    else if (data <= -200)
    {
        //kiri
        flag = 2;
        kiri();
        analogWrite(pwmA, 125 - P);
        analogWrite(pwmB, 125 - P);
    }
}

void parsingData()
{
    int j=0;

```

```

dt[j]="";
//proses parsing data
for(i=1;i<dataIn.length();i++)
{
    //pengecekan tiap karakter dengan karakter (#) dan (,)
    if ((dataIn[i] == '#') || (dataIn[i] == ','))
    {
        //increment variabel j, digunakan untuk merubah
index array penampung
        j++;
        dt[j]="";          //inisialisasi variabel array
dt[j]
    }
    else
    {
        //proses tampung data saat pengecekan karakter
selesai.
        dt[j] = dt[j] + dataIn[i];
    }
}
//kirim data hasil parsing
data = dt[0].toInt();
luas = dt[1].toInt();
}

```

D. Program Image

```
#from libraryshape import ShapeDetector
from collections import deque
import argparse
import cv2
import time
import threading
import time
import threading
import math
import numpy as np
import serial
import imutils
import freenect
tinggi=5

global counter
global dX
global dY
global X
global Y
global error
global direction
global resolusi
global counter
global img

ser = serial.Serial(
    port='/dev/ttyACM0',
    baudrate=9600,
    parity=serial.PARITY_NONE,
    stopbits=serial.STOPBITS_ONE,
)
ser.isOpen()
def kontrol():
    global out
    out = ''
    while True :
        while ser.inWaiting() > 0:
            out = ser.read(3)
def image():
    global tanda
    global errorX
    tanda =0
    errorX=0
    def get_video():
        array,_ = freenect.sync_get_video()
        array = cv2.cvtColor(array,cv2.COLOR_RGB2BGR)
        return array
```

```

def nothing(x):
    pass

ap = argparse.ArgumentParser()
ap.add_argument("-v", "--video",
    help="path to the (optional) video file")
ap.add_argument("-b", "--buffer", type=int, default=32,
    help="max buffer size")
args = vars(ap.parse_args())
cv2.namedWindow('Frame')
cv2.namedWindow('track')
cv2.resizeWindow('track', 500,250)
cv2.createTrackbar('VminH', 'track', 29, 180, nothing)
cv2.createTrackbar('VmaxH', 'track', 84, 180, nothing)
cv2.createTrackbar('VminS', 'track', 86, 255, nothing)
cv2.createTrackbar('VmaxS', 'track', 255, 255, nothing)
cv2.createTrackbar('VminV', 'track', 6, 255, nothing)
cv2.createTrackbar('VmaxV', 'track', 255, 255, nothing)
pts = deque(maxlen=args["buffer"])
counter = 0
(dx, dy) = (0, 0)
(X,Y) = (0,0)
(error) = 999999999
direction = ""
resolusi=620
counter=1
tanda = 0
errorX = 0

while True:
    img = get_video()
    img = cv2.flip(img, 2)
    frame = img.copy()
    frame = imutils.resize(frame, width=resolusi)
    img = imutils.resize(img, width=resolusi)
    if args.get("video") and not grabbed:
        break
    blurred = cv2.GaussianBlur(frame, (11, 11), 0)
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    VminH = cv2.getTrackbarPos('VminH', 'track')
    VmaxH = cv2.getTrackbarPos('VmaxH', 'track')
    VminS = cv2.getTrackbarPos('VminS', 'track')
    VmaxS = cv2.getTrackbarPos('VmaxS', 'track')
    VminV = cv2.getTrackbarPos('VminV', 'track')
    VmaxV = cv2.getTrackbarPos('VmaxV', 'track')
    lower = np.array([VminH, VminS, VminV], np.uint8)
    upper = np.array([VmaxH, VmaxS, VmaxV], np.uint8)
    frame = cv2.inRange(hsv, lower, upper)
    frame = cv2.erode(frame, None, iterations=2)
    frame = cv2.dilate(frame, None, iterations=2)

```



```

cnts = cv2.findContours(frame.copy(),
cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)[-2]
center = None
(X, Y) = (0, 0)
if len(cnts) > 0:
    c = max(cnts, key=cv2.contourArea)
    ((x, y), radius) = cv2.minEnclosingCircle(c)
    M = cv2.moments(c)
    center = (int(M["m10"] / M["m00"]), int(M["m01"] /
M["m00"]))
    X = int(x)
    Y = int(y)
    errorX = (resolusi/2- X)
    errorY = (resolusi/4+resolusi/8 - Y)
    errorXY = ((errorX * errorX) + (errorY * errorY))
    error = math.sqrt(errorXY)
    print "R=", radius , " Error=" , errorX
    ser.write('*'+str(errorX)+'*'+str(radius)+"#\r\n")
    if radius > 3:
        cv2.circle(img, (int(x), int(y)), int(radius),
(0, 255, 255), 2)
        cv2.circle(img, center, 5, (0, 0, 255), -1)
        pts.appendleft(center)
    if radius > 170:
        print "STOP"
else :
    print "x"
    ser.write('*'+ "999"+',' + str(radius) + "#\r\n")
    for i in np.arange(1, len(pts)):
        img9 = cv2.line (img, (resolusi/2, resolusi /
4+resolusi/8), (X, Y), (255, 255, 0), 3)
        if counter >= 10 and i == 10 and pts[i-10] is not
None:
            dX = pts[i-10][0] - pts[i][0]
            dY = pts[i-10][1] - pts[i][1]
            (dirX, dirY) = ("", "")
            if np.abs(dX) > 20:
                dirX = "Kanan Gan" if np.sign(dX) == 1 else
"Kiri Gan"
            if np.abs(dY) > 20:
                dirY = "Bawah" if np.sign(dY) == 1 else
"Atas"
            if dirX != "" and dirY != "":
                direction = "{}-{}".format(dirY, dirX)
            else:
                direction = dirX if dirX != "" else dirY
            thickness = int(np.sqrt(args["buffer"] / float(i +
1)) * 2.5)
            cv2.line(img, pts[i - 1], pts[i], (0, 0, 255),
thickness)

```

```

        cv2.putText(img, "{}".format(out), (270, 270),
cv2.FONT_HERSHEY_SIMPLEX, 2, (0, 0, 255), 5)
cv2.putText(img, "G A S".format(out), (280, 200),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
cv2.putText(img, direction, (10, 30),
cv2.FONT_HERSHEY_SIMPLEX,
0.65, (0, 0, 255), 3)
cv2.putText(img, "dx: {}, dy: {}".format(dx, dy),
(10, img.shape[0] - 10), cv2.FONT_HERSHEY_SIMPLEX,
0.35, (0, 0, 255), 1)
cv2.putText(img, "X: {}, Y: {}".format(X, Y),
(130, img.shape[0] - 10), cv2.FONT_HERSHEY_SIMPLEX,
0.35, (0, 0, 255), 1)
cv2.putText(img, "ERROR : {}".format(error),
(230, img.shape[0] - 10),
cv2.FONT_HERSHEY_SIMPLEX,
0.35, (0, 0, 255), 1)
cv2.imshow('Original', img)
cv2.imshow("Frame", frame)
key = cv2.waitKey(1) & 0xFF
if key== ord('r'):
    counter-=1
    if counter <=0:
        counter=0
    print counter
if key== ord('n'):
    counter+=1
    if counter >=10:
        counter=10
    print counter
if key == ord("q"):
    break
camera.release()
cv2.destroyAllWindows()
kontrol = threading.Thread(name='kontrol',
target=kontrol,args=())
image = threading.Thread(name='image', target=image,args=())
image.start()

kontrol.start()

```

-----Halaman ini sengaja dikosongkan-----

BIODATA PENULIS



Muhammad Iqbal Fahmi lahir di Brebes, 15 Januari 1995, yang merupakan anak dari pasangan Badaturohman dan Istri Anita Triana. Penulis menyelesaikan pendidikan dasar di SD Islam Ta'allumul Huda Bumiayu (2001 s.d. 2007) lalu dilanjutkan dengan pendidikan menengah di SMP Islam Ta'allumul Huda Bumiayu (2007 s.d. 2010) dan SMK N1 Tonjong (2010 s.d. 2013). Penulis mengenyam pendidikan tahap D3 di jurusan Teknik Elektro, Sekolah Vokasi, Universitas Gadjah Mada yang kemudian melanjutkan ke jenjang S-1 di jurusan

Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya.

Email : ifahmi36@gmail.com

HP/WA : 0896-7070-8262